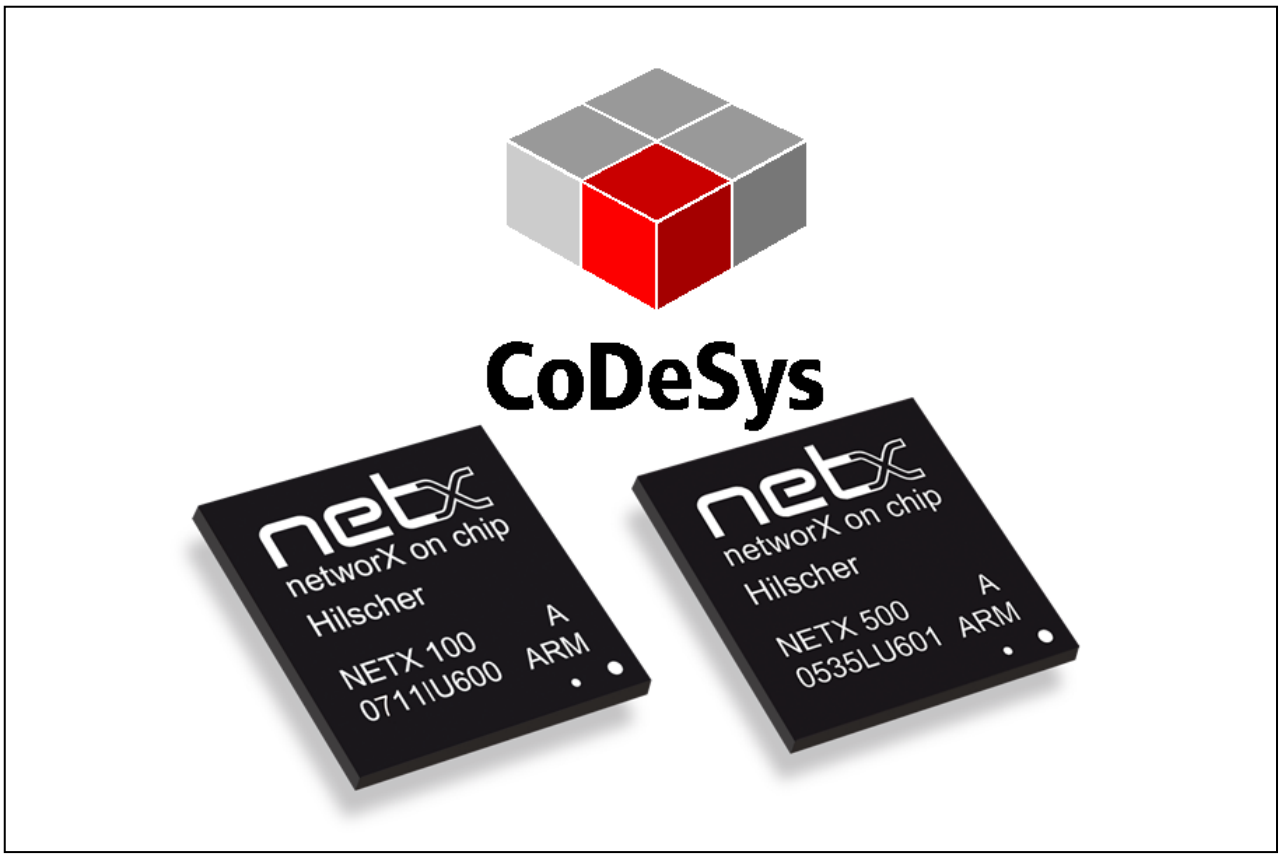
Benutzerhandbuch

# netPLC mit CoDeSys

## Inbetriebnahme

**V3.5**




**Hilscher Gesellschaft für Systemautomation mbH**

**www.hilscher.com**

DOC120113UM03DE | Revision 3 | Deutsch | 2014-07 | Freigegeben | Öffentlich

# Inhaltsverzeichnis

1 EINLEITUNG ..... 5
1.1 Über das Benutzerhandbuch ..... 5
1.1.1 Änderungsübersicht ..... 5
1.1.2 Konventionen in diesem Handbuch ..... 6
1.2 netPLC mit CoDeSys ..... 7
1.2.1 Bezug auf Hardware, Firmware, Server, Treiber und Software ..... 7
1.2.2 Dokumentationen netPLC ..... 8
1.3 Rechtliche Hinweise ..... 9
1.3.1 Copyright ..... 9
1.3.2 Wichtige Hinweise ..... 9
1.3.3 Haftungsausschluss ..... 10
1.3.4 Gewährleistung ..... 10
1.3.5 Exportbestimmungen ..... 11
1.3.6 Eingetragene Warenzeichen ..... 11
1.4 Lizenzen ..... 11

2 SICHERHEIT ..... 12
2.1 Sicherheitshinweise ..... 12
2.2 Bestimmungsgemäßer Gebrauch ..... 12
2.3 Personalqualifizierung ..... 12
2.4 Pflicht zum Lesen des Benutzerhandbuches ..... 12

3 KURZBESCHREIBUNG UND VORAUSSETZUNGEN ..... 13
3.1 Kurzbeschreibung ..... 13
3.2 Systemvoraussetzungen ..... 13
3.2.1 Systemvoraussetzungen PC ..... 13
3.2.2 Systemvoraussetzungen CoDeSys Programmiersystem ..... 13

4 DAS ERSTE PROJEKT FÜR DIE NETPLC ERSTELLEN (BEISPIEL NPLC-C100) ..... 14
4.1 Übersicht ..... 14
4.2 Vorbereitung ..... 14
4.2.1 netPLC CoDeSys PC-Karten Server ..... 14
4.2.2 CoDeSys starten ..... 15
4.3 Projekt anlegen, Gerät und Programmierart auswählen ..... 17
4.3.1 Host E/As bei NPLC-C100 Slot-SPS verwenden ..... 19
4.3.2 Lokale E/As beim NPLC-M100 Modul verwenden ..... 21
4.4 Buskonfiguration erstellen ..... 25
4.4.1 Feldbussystem für die Steuerung auswählen ..... 25
4.4.2 PROFIBUS DP Master konfigurieren ..... 27
4.4.3 CANopen Master Konfigurationsfenster einfügen ..... 28
4.4.4 DeviceNet Scanner konfigurieren ..... 32
4.5 Busaufbau abbilden ..... 36
4.5.1 Slave Gerätedateien importieren ..... 36
4.5.2 Slave-Gerät in den Busaufbau anhängen und konfigurieren ..... 37
4.5.3 Adressen Ein-/Ausgänge des Slaves ermitteln ..... 39
4.6 Steuerungsprogramm erstellen ..... 40


© Hilscher, 2009 - 2014

4.6.1 Symbolkonfiguration erstellen ....43
4.6.2 Projekt übersetzen....47
4.7 Verbindung zur Steuerung herstellen ....48
4.7.1 Gateway festlegen....48
4.7.2 Netzwerk durchsuchen und Aktiven Pfad setzen....49
4.8 Steuerungsprogramm und Buskonfiguration übertragen....51
4.8.1 Einloggen und übertragen....51
4.8.2 Bootprojekt erzeugen....53
4.8.3 Bootprojekt entfernen....53
4.8.4 Steuerungsprogramm starten....54
4.9 Verbindung zur Steuerung aufheben....55

5 VISUALISIERUNG....56
5.1 Visualisierung HMI-Steuerung netHMI PLC....56
5.2 Visualisierung Slot-SPS NPLC-C100 und Modul-SPS NPLC-M100....56
5.2.1 OPC Konfigurator....57
5.2.2 OPC-Klient....61

6 HOST-E/A-SCHNITTSTELLE (SLOT-SPS NPLC-C100)....62

7 FUNKTIONEN....63
7.1 Funktionsbibliothek erweitern....63
7.2 Funktionen – HilPlcSysFunctions....64
7.2.1 Batterie Status – StatusOfBattery....66
7.2.2 Status externe Spannungsversorgung – StatusOfExt24VDC....67
7.2.3 Status Spannungsversorgung über PCI – StatusOfPCIPower....68
7.3 Reset-Verhalten (NPLC-C100)....69
7.4 RUN/STOP-Betriebswahlschalter (NPLC-C100 und NPLC-M100)....69
7.5 Beispiel SPS-Programm zur Batterie-Überwachung (NPLC-C100 und NPLC-M100)....70

8 BUS DIAGNOSE....71
8.1 Bus in Betrieb nehmen und testen....71
8.1.1 Einloggen und Programm übertragen....71
8.1.2 Gerätezuordnung....72
8.1.3 Eingangs- und Ausgangsdaten prüfen....73

9 FEHLERSUCHE....74
9.1 Fehlermeldung bei der Bootapplikationserzeugung....74

10 LEDS....75
10.1 Übersicht LEDs NPLC-C100 und NPLC-M100....75
10.2 Übersicht LEDs NHMI J500Q5-DP-DP/DPM/CDS/QV (PROFIBUS DP Master)....75
10.3 Übersicht LEDs NHMI J500Q5-RE-DP/PNM/CDS/QV (PROFINET IO Controller)....76
10.4 LED SYS....76
10.5 LED APL....77
10.6 LED CANopen Master....77
10.7 LED DeviceNet Master....77
10.8 LED PROFIBUS DP-Master....78


© Hilscher, 2009 - 2014

10.9 LEDs PROFINET IO Controller ....78

11 FEHLERNUMMERN ....79
11.1 Fehlernummern Funktionen HilPlcSysFunctions ....79

12 TECHNISCHE DATEN DER PROTOKOLLE ....80
12.1 CANopen Master und DeviceNet Master ....80
12.2 PROFIBUS DP-Master ....82
12.2.1 NPLC-C100-DP und NPLC M100-DP ....82
12.2.2 NHMI J500Q5-DP-DP ....83
12.3 PROFINET IO-RT-Controller ....84

13 ANHANG ....85
13.1 netPLC CoDeSys Server Programm ....85
13.1.1 netPLC CoDeSys Server starten ....85
13.1.2 netPLC CoDeSys Server Fenster ....86
13.1.3 netPLC CoDeSys Server beenden ....87
13.2 Beispielprojekte ....88
13.2.1 netPLC CoDeSys Produkt-DVD ....88
13.2.2 netHMI PLC Produkt-DVD ....90
13.3 Auslastungsinformationen ....91
13.4 Abbildungsverzeichnis ....92
13.5 Tabellenverzeichnis ....93
13.6 Kontakte ....94


© Hilscher, 2009 - 2014

# 1 Einleitung

## 1.1 Über das Benutzerhandbuch

Dieses Benutzerhandbuch beschreibt die Inbetriebnahme von netPLC-Komponenten mit der SPS-Programmiersoftware CoDeSys.

Zu netPLC-Komponenten gehören

- Slot-SPS **NPLC-C100**
- Modul-SPS **NPLC-M100**
- HMI-Steuerung netHMI PLC **NHMI J500Q5-DP-DP/DPM/CDS/QV**
- HMI-Steuerung netHMI PLC **NHMI J500Q5-RE-DP/PNM/CDS/QV**

### 1.1.1 Änderungsübersicht

| Index | Datum | Kapitel | Änderungen |
|---|---|---|---|
| 1 | 21.05.2012 | Alle | erstellt |
| 2 | 27.01.2014 | 1.2.1 | Abschnitt *Bezug auf Hardware, Firmware, Server, Treiber und Software* aktualisiert. |
| | | 4 | Kapitel *Das erste Projekt für die netPLC erstellen (Beispiel NPLC-C100)* überarbeitet. |
| | | 4.3.1 | Abschnitt *Host E/As bei NPLC-C100 Slot-SPS* ergänzt. |
| | | 6 | Kapitel *Host-E/A-Schnittstelle (Slot-SPS NPLC-C100)* ergänzt. |
| | | 12.1 | Technische Daten für *CANopen Master und DeviceNet Master* zusammengefasst und aktualisiert. |
| | | 12.2 | Technische Daten für *PROFIBUS DP-Master* aktualisiert und DPV1-Funktionen ergänzt. |
| | | 13.2.1.5 | Abschnitt zu Beispiel *NPLC-C100-DP with Host IOs* ergänzt. |
| | | 13.2.1.6 | Abschnitt zu Beispiel *NPLC-C100-DP DPV1* ergänzt. |
| 3 | 09.07.2014 | Alle | Neue netPLC-Komponente **netHMI PLC** in Dokument ergänzt. |
| | | 10.2 | Abschnitt *Übersicht LEDs NHMI J500Q5-DP-DP/DPM/CDS/QV (PROFIBUS DP Master)* eingefügt. |
| | | 10.3 | Abschnitt *Übersicht LEDs NHMI J500Q5-RE-DP/PNM/CDS/QV (PROFINET IO Controller)* eingefügt. |
| | | 10.9 | Abschnitt *LEDs PROFINET IO Controller* eingefügt. |
| | | 12.2 | Technische Daten für *PROFIBUS DP-Master* aktualisiert. |
| | | 12.3 | Technische Daten für *PROFINET IO-RT-Controller* eingefügt. |

*Tabelle 1: Änderungsübersicht*


© Hilscher, 2009 - 2014

## 1.1.2 Konventionen in diesem Handbuch

Hinweise, Handlungsanweisungen und Ergebnisse von Handlungen sind wie folgt gekennzeichnet:

**Hinweise**

**Wichtig:** <Wichtiger Hinweis>

**Hinweis:** <Hinweis>

<Hinweis, wo Sie weitere Informationen finden können>

**Handlungsanweisungen**

1. <Anweisung>
2. <Anweisung>

oder

➢ <Anweisung>

**Ergebnisse**

<Ergebnis>


© Hilscher, 2009 - 2014

# 1.2 netPLC mit CoDeSys

## 1.2.1 Bezug auf Hardware, Firmware, Server, Treiber und Software

**Hinweis:** Die aufgeführten Hardware-Revisionen, Firmware- und Treiberversionen bzw. die Versionen für die Programmiersoftware CoDeSys gehören funktional zusammen.

### netPLC-Komponenten (Hardware)

| netPLC-Komponenten | Artikel Nr. | Revision |
|---|---|---|
| NPLC-C100-DP/CDS-OPC | 1800.410/CDS-OPC | 4 |
| NPLC-C100-CO/CDS-OPC | 1800.500/CDS-OPC | 4 |
| NPLC-C100-DN/CDS-OPC | 1800.510/CDS-OPC | 4 |
| NPLC-M100-DP/CDS-OPC | 1830.410/CDS-OPC | 3 |
| NHMI J500Q5-DP-DP/DPM/CDS/QV | 1878.180/DPM/CDS/QV | 3 |
| NHMI J500Q5-RE-DP/PNM/CDS/QV | 1872.180/PNM/CDS/QV | 1 |

*Tabelle 2: Bezug auf netPLC-Komponenten (Hardware)*

### Firmware

| Firmware-Datei | Gerätetyp | Feldbussystem | Firmware-Version |
|---|---|---|---|
| SC111000.NXF | NPLC-C100-DP | PROFIBUS DP Master | 4.0 |
| SC116000.NXF | NPLC-C100-CO | CANopen Master | 4.0 |
| | NPLC-C100-DN | DeviceNet Master | 4.0 |
| SM211000.NXF | NPLC M100-DP | PROFIBUS DP Master | 4.0 |
| HJ111400.NXF | NHMI J500Q5-DP-DP | PROFIBUS DP Master | 4.0 |
| HJ11C400.NXF | NHMI J500Q5-RE-DP | PROFINET IO Controller | 4.0 |

*Tabelle 3: Bezug auf Firmware*

### Treiber

| Treiber | Treiberversion |
|---|---|
| cifX Device Driver | 1.2.x.x |
| USB Treiber | 5.1.2600.2180 |

*Tabelle 4: Bezug auf Treiber*

### Server und TCP/IP Gateway

| Software | Software-Version |
|---|---|
| CoDeSys Gateway | 3.5.0.0 |
| CoDeSys Service Control | 3.5.0.0 |
| CoDeSys Control Win V3 | 3.5.0.0 |

*Tabelle 5: Bezug auf Server*

### Software

| Software | Software-Version |
|---|---|
| CoDeSys | V3.5 |

*Tabelle 6: Bezug auf Software*


© Hilscher, 2009 - 2014

### 1.2.2 Dokumentationen netPLC

Die nachfolgende Dokumentationsübersicht gibt Auskunft darüber, in welchem Handbuch Sie zu welchen Inhalten weitere Informationen finden können.

Alle in der Übersicht aufgeführten Handbücher sind auf der mitgelieferten DVD unter dem Verzeichnis **Documentation** als PDF-Datei zu finden.

| **Handbuch** | **Titel**<br>**Inhalt** | **Dateiname** |
|---|---|---|
| Benutzerhandbuch | netPLC mit CoDeSys, Software-Installation<br>Inhalt der Produkt-DVD<br>Voraussetzungen<br>Systemübersicht<br>Software installieren<br>Technische Daten der Protokolle | netPLC mit CoDeSys – Software-Installation UM xx DE.pdf |
| Benutzerhandbuch | Slot-SPS NPLC-C100, Hardware-Installation<br>Voraussetzungen<br>Gerätezeichnungen<br>Hardware installieren<br>Batterie<br>Hardwarebeschreibung<br>Technische Daten Hardware<br><br>Anmerkung: Die Beschreibung der LEDs ist im Dokument ‚netPLC mit CoDeSys, Inbetriebnahme' enthalten. | Slot-SPS NPLC-C100 - Hardware-Installation UM xx DE.pdf |
| Benutzerhandbuch | netPLC mit CoDeSys, Inbetriebnahme<br>Das erste Projekt für die netPLC erstellen<br>Buskonfiguration erstellen<br>Steuerungsprogramm erstellen<br>Verbindung zur Steuerung herstellen<br>Steuerungsprogramm und Buskonfiguration übertragen<br>Verbindung zur Steuerung aufheben<br>Visualisierung<br>Funktionen<br>Diagnose<br>Fehlersuche<br>Beschreibung der LEDs | netPLC mit CoDeSys - Inbetriebnahme UM xx DE.pdf<br>(dieses Dokument) |
| Anwenderdokumentation | CoDeSys V3, Installation und Erste Schritte | CoDeSys Installation und Erste Schritte.pdf |
| Anwenderdokumentation | CoDeSys OPC Server 3 Installation und Benutzung | OPC_V3_how_to_use_D.pdf |
| Benutzerhandbuch | Verkabelungshinweise, PROFIBUS, CANopen, DeviceNet, AS-Interface, CompoNet, CC-Link<br>Angaben zu Kabeleigenschaften, max. Kabellängen in Abhängigkeit der Baudrate sowie Abschlusswiderständen. | Verkabelungshinweise UM xx DE.pdf |

*Tabelle 7: Dokumentationen netPLC CoDeSys*

**Hinweis:** Nach der Installation des CoDeSys Programmiersystems finden Sie weitere Dokumente im Installationsverzeichnis, typischerweise unter: **C:\Programme\3S CoDeSys\CoDeSys\Documentation**


© Hilscher, 2009 - 2014

# 1.3 Rechtliche Hinweise

## 1.3.1 Copyright

© Hilscher, 2009 - 2014, Hilscher Gesellschaft für Systemautomation mbH

Alle Rechte vorbehalten.

Die Bilder, Fotografien und Texte der Begleitmaterialien (Benutzerhandbuch, Begleittexte, Dokumentation etc.) sind durch deutsches und internationales Urheberrecht sowie internationale Handels- und Schutzbestimmungen geschützt. Sie sind ohne vorherige schriftliche Genehmigung nicht berechtigt, diese vollständig oder teilweise durch technische oder mechanische Verfahren zu vervielfältigen (Druck, Fotokopie oder anderes Verfahren), unter Verwendung elektronischer Systeme zu verarbeiten oder zu übertragen. Es ist Ihnen untersagt, Veränderungen an Copyrightvermerken, Kennzeichen, Markenzeichen oder Eigentumsangaben vorzunehmen. Darstellungen werden ohne Rücksicht auf die Patentlage mitgeteilt. Die in diesem Dokument enthaltenen Firmennamen und Produktbezeichnungen sind möglicherweise Marken bzw. Warenzeichen der jeweiligen Inhaber und können warenzeichen-, marken- oder patentrechtlich geschützt sein. Jede Form der weiteren Nutzung bedarf der ausdrücklichen Genehmigung durch den jeweiligen Inhaber der Rechte.

## 1.3.2 Wichtige Hinweise

Das Benutzerhandbuch, Begleittexte und die Dokumentation wurden mit größter Sorgfalt erarbeitet. Fehler können jedoch nicht ausgeschlossen werden. Eine Garantie, die juristische Verantwortung für fehlerhafte Angaben oder irgendeine Haftung kann daher nicht übernommen werden. Sie werden darauf hingewiesen, dass Beschreibungen in dem Benutzerhandbuch, den Begleittexte und der Dokumentation weder eine Garantie, noch eine Angabe über die nach dem Vertrag vorausgesetzte Verwendung oder eine zugesicherte Eigenschaft darstellen. Es kann nicht ausgeschlossen werden, dass das Benutzerhandbuch, die Begleittexte und die Dokumentation nicht vollständig mit den beschriebenen Eigenschaften, Normen oder sonstigen Daten der gelieferten Produkte übereinstimmen. Eine Gewähr oder Garantie bezüglich der Richtigkeit oder Genauigkeit der Informationen wird nicht übernommen.

Wir behalten uns das Recht vor, unsere Produkte und deren Spezifikation, sowie zugehörige Benutzerhandbücher, Begleittexte und Dokumentationen jederzeit und ohne Vorankündigung zu ändern, ohne zur Anzeige der Änderung verpflichtet zu sein. Änderungen werden in zukünftigen Manuals berücksichtigt und stellen keine Verpflichtung dar; insbesondere besteht kein Anspruch auf Überarbeitung gelieferter Dokumente. Es gilt jeweils das Manual, das mit dem Produkt ausgeliefert wird.

Die Hilscher Gesellschaft für Systemautomation mbH haftet unter keinen Umständen für direkte, indirekte, Neben- oder Folgeschäden oder Einkommensverluste, die aus der Verwendung der hier enthaltenen Informationen entstehen.

### 1.3.3 Haftungsausschluss

Die Software wurde von der Hilscher Gesellschaft für Systemautomation mbH sorgfältig erstellt und getestet und wird im reinen Ist-Zustand zur Verfügung gestellt. Es kann keine Gewährleistung für die Leistungsfähigkeit und Fehlerfreiheit der Software für alle Anwendungsbedingungen und -fälle und die erzielten Arbeitsergebnisse bei Verwendung der Software durch den Benutzer übernommen werden. Die Haftung für etwaige Schäden, die durch die Verwendung der Hard- und Software oder der zugehörigen Dokumente entstanden sein könnten, beschränkt sich auf den Fall des Vorsatzes oder der grob fahrlässigen Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen vorhersehbaren Schaden begrenzt.

Es ist strikt untersagt, die Software in folgenden Bereichen zu verwenden:

- für militärische Zwecke oder in Waffensystemen;
- zum Entwurf, zur Konstruktion, Wartung oder zum Betrieb von Nuklearanlagen;
- in Flugsicherungssystemen, Flugverkehrs- oder Flugkommunikationssystemen;
- in Lebenserhaltungssystemen;
- in Systemen, in denen Fehlfunktionen der Software körperliche Schäden oder Verletzungen mit Todesfolge nach sich ziehen können.

Sie werden darauf hingewiesen, dass die Software nicht für die Verwendung in Gefahrumgebungen erstellt worden ist, die ausfallsichere Kontrollmechanismen erfordern. Die Benutzung der Software in einer solchen Umgebung geschieht auf eigene Gefahr; jede Haftung für Schäden oder Verluste aufgrund unerlaubter Benutzung ist ausgeschlossen.

### 1.3.4 Gewährleistung

Obwohl die Hard- und Software mit aller Sorgfalt entwickelt und intensiv getestet wurde, übernimmt die Hilscher Gesellschaft für Systemautomation mbH keine Garantie für die Eignung für irgendeinen Zweck, der nicht schriftlich bestätigt wurde. Es kann nicht gewährleistet werden, dass die Hard- und Software Ihren Anforderungen entspricht, die Verwendung der Software unterbrechungsfrei und die Software fehlerfrei ist. Eine Garantie auf Nichtübertretung, Nichtverletzung von Patenten, Eigentumsrecht oder Freiheit von Einwirkungen Dritter wird nicht gewährt. Weitere Garantien oder Zusicherungen hinsichtlich Marktgängigkeit, Rechtsmängelfreiheit, Integrierung oder Brauchbarkeit für bestimmte Zwecke werden nicht gewährt, es sei denn, diese sind nach geltendem Recht vorgeschrieben und können nicht eingeschränkt werden. Gewährleistungsansprüche beschränken sich auf das Recht, Nachbesserung zu verlangen.


© Hilscher, 2009 - 2014

### 1.3.5 Exportbestimmungen

Das gelieferte Produkt (einschließlich der technischen Daten) unterliegt den gesetzlichen Export- bzw. Importgesetzen sowie damit verbundenen Vorschriften verschiedener Länder, insbesondere denen von Deutschland und den USA. Die Software darf nicht in Länder exportiert werden, in denen dies durch das US-amerikanische Exportkontrollgesetz und dessen ergänzender Bestimmungen verboten ist. Sie verpflichten sich, die Vorschriften strikt zu befolgen und in eigener Verantwortung einzuhalten. Sie werden darauf hingewiesen, dass Sie zum Export, zur Wiederausfuhr oder zum Import des Produktes unter Umständen staatlicher Genehmigungen bedürfen.

### 1.3.6 Eingetragene Warenzeichen

CANopen® ist eine registrierte Warenmarke des CAN in AUTOMATION - International Users and Manufacturers Group e.V., Nürnberg.

CoDeSys® ist eine registrierte Warenmarke der 3S-Smart Software Solutions GmbH, Kempten

DeviceNet™ und EtherNet/IP™ sind Warenmarken der ODVA (Open DeviceNet Vendor Association, Inc).

PROFIBUS® und PROFINET® sind registrierte Warenmarken von PROFIBUS International, Karlsruhe.

Windows® XP, Windows® 7 und Windows® 8 sind registrierte Warenmarken der Microsoft Corporation.

Alle anderen erwähnten Marken sind Eigentum ihrer jeweiligen rechtmäßigen Inhaber.

## 1.4 Lizenzen

Für den Betrieb der netPLC-Komponente mit CoDeSys SPS-Funktion ist eine CoDeSys-Lizenz nötig. Diese Lizenz ist in der netPLC Geräte-Hardware gespeichert und im Lieferumfang enthalten. Sie erlaubt die Nutzung und den Betrieb des CoDeSys Programmiersystems und der SPS-Laufzeitumgebung in der Gerätehardware.

Eine Lizenz kann in der netPLC Gerätehardware nicht gelöscht oder umgewandelt werden.

Für den Betrieb als Feldbus-Master ist in der netPLC Gerätehardware eine Master-Lizenz gespeichert. Diese gewährt die Nutzung des jeweiligen Feldbussystems als Master.


© Hilscher, 2009 - 2014

# 2 Sicherheit

## 2.1 Sicherheitshinweise

Das Benutzerhandbuch, die Begleittexte und die Dokumentation sind für die Verwendung der Produkte durch ausgebildetes Fachpersonal erstellt worden. Bei der Nutzung der Produkte sind sämtliche Sicherheitshinweise sowie alle geltenden Vorschriften zu beachten. Technische Kenntnisse werden vorausgesetzt. Der Verwender hat die Einhaltung der Gesetzesbestimmungen sicherzustellen.

## 2.2 Bestimmungsgemäßer Gebrauch

Die in diesem Benutzerhandbuch beschriebenen netPLC-Komponenten besitzen SPS-Funktionalität mit unterlagerter Feldbus-Kommunikationsschnittstelle.

Zu netPLC-Komponenten gehören

- Slot-SPS **NPLC-C100**
- Modul-SPS **NPLC-M100**
- HMI-Steuerung netHMI PLC **NHMI J500Q5-DP-DP/DPM/CDS/QV**
- HMI-Steuerung netHMI PLC **NHMI J500Q5-DP-DP/PNM/CDS/QV**

Abhängig von der verwendeten netPLC-Komponente können die in Abschnitt *Bezug auf Hardware, Firmware, Server, Treiber und Software* (Seite 7) aufgeführten Feldbussysteme automatisiert werden.

## 2.3 Personalqualifizierung

Die netPLC-Komponente darf ausschließlich von qualifiziertem Fachpersonal montiert, konfiguriert, betrieben oder demontiert werden.

## 2.4 Pflicht zum Lesen des Benutzerhandbuches

**Wichtig:** Vor der Installation und Verwendung Ihres Gerätes müssen Sie alle Instruktionen in diesem Handbuch lesen und verstehen, um Schaden zu vermeiden.


© Hilscher, 2009 - 2014

# 3 Kurzbeschreibung und Voraussetzungen

## 3.1 Kurzbeschreibung

CoDeSys ist ein Steuerungsprogrammiersystem. In Übereinstimmung mit der Norm IEC 61131-3 unterstützt es alle Standard-IEC-Programmiersprachen. Dieses Benutzerhandbuch beschreibt die geräteabhängige Programmierung für netPLC-Komponenten, inklusive der speziell für die netPLC-Komponenten in CoDeSys implementierten Überwachungsfunktionen und deren Anwendung.

Zu netPLC-Komponenten gehören

- Slot-SPS **NPLC-C100**
- Modul-SPS **NPLC-M100**
- HMI-Steuerung netHMI PLC **NHMI J500Q5-DP-DP/DPM/CDS/QV**
- HMI-Steuerung netHMI PLC **NHMI J500Q5-RE-DP/PNM/CDS/QV**

## 3.2 Systemvoraussetzungen

### 3.2.1 Systemvoraussetzungen PC

- PC mit mind. 1,8 GHz Prozessor empfohlen 3,0 GHz.
- Windows® XP oder Windows® 7
- DVD-ROM-Laufwerk
- Grafikauflösung: mind. 1024 x 768 Pixel oder mehr
- Tastatur und Maus
- Für Slot-SPS NPLC-C100: PC mit PCI-Anschluss mit 3,3 V bzw. 5 V für Hardware-Revision 4 der NPLC-C100-xx

### 3.2.2 Systemvoraussetzungen CoDeSys Programmiersystem

Systemvoraussetzungen für das CoDeSys Programmiersystem

- PC mit mind. 1,8 GHz Prozessor empfohlen 3,0 GHz.
- Windows® XP oder Windows® 7
- .NET Framework 4.0 (wird sonst mitinstalliert, Internetzugang erforderlich).
- Freier Festplattenspeicher: mind. 600 MByte, empfohlen 1 GByte
- RAM: mind. 512 MB, empfohlen 1024 MByte
- Auflösung: mind. mind. 1024 x 768 Bildpunkte
- Tastatur und Maus


© Hilscher, 2009 - 2014

# 4 Das erste Projekt für die netPLC erstellen (Beispiel NPLC-C100)

## 4.1 Übersicht

Dieses Kapitel beschreibt, wie ein Projekt für eine netPLC-Komponente mit dem CoDeSys Programmiersystem erstellt wird.

In dem dargestellten Beispiel wird exemplarisch eine NPLC-C100-DP verwendet. Für den Betrieb wird mindestens ein PROFIBUS DP Slave und ein PROFIBUS-Kabel benötigt.

Grundsätzlich sind die nachfolgend beschriebenen Vorgehensweisen auch auf die anderen netPLC-Komponenten anwendbar. Sollten bei den anderen netPLC-Komponenten Unterschiede bestehen, wird im jeweiligen Abschnitt explizit darauf hingewiesen.

Wenn das CoDeSys-Projekt über den USB-Anschluss der netPLC-Komponente geladen werden soll, wird noch ein USB-Kabel vom Typ A auf Typ mini B benötigt. Soll das Projekt über Ethernet geladen werden, benötigen Sie eine Ethernet-Verbindung zu dem Gerät.

## 4.2 Vorbereitung

### 4.2.1 netPLC CoDeSys PC-Karten Server

Der CoDeSys PC-Karten Server ist nur erforderlich und wird nur installiert, wenn mit einer Slot-SPS NPLC-C100 über den PCI-Bus eines PCs mit der Slot-SPS kommuniziert wird. Im Benutzerhandbuch *netPLC mit CoDeSys, Software-Installation* beschreibt das Kapitel *Systemübersicht*, wann der Server benötigt wird.

Der netPLC CoDeSys PC-Karten Server wird in Windows so installiert, dass er automatisch mit jedem Start des PCs gestartet wird.

**Hinweis:** Stellen Sie sicher, dass der Server läuft, wenn mit der Slot-SPS NPLC-C100 über den PCI-Bus des PCs kommuniziert wird.

Prüfen, ob der netPLC CoDeSys Server läuft.

➢ Kontrollieren Sie, ob das Symbol im Systemtray erscheint.

➢ Wenn der netPLC CoDeSys Server noch nicht gestartet wurde, dann starten Sie diesen jetzt über **Start > Alle Programme > Hilscher GmbH > netPLC CoDeSys Server**.

↬ Im Tray wird das Symbol angezeigt.


© Hilscher, 2009 - 2014

### 4.2.2 CoDeSys starten

CoDeSys starten.

- Wählen Sie **Start > Alle Programme > 3S CoDeSys > CoDeSys > CoDeSys V3.5** oder
- Starten Sie CoDeSys mit einem Doppelklick auf das Desktop-Icon

CoDeSys wird gestartet.

© Hilscher, 2009 - 2014

➪ Die CoDeSys Programmieroberfläche erscheint.

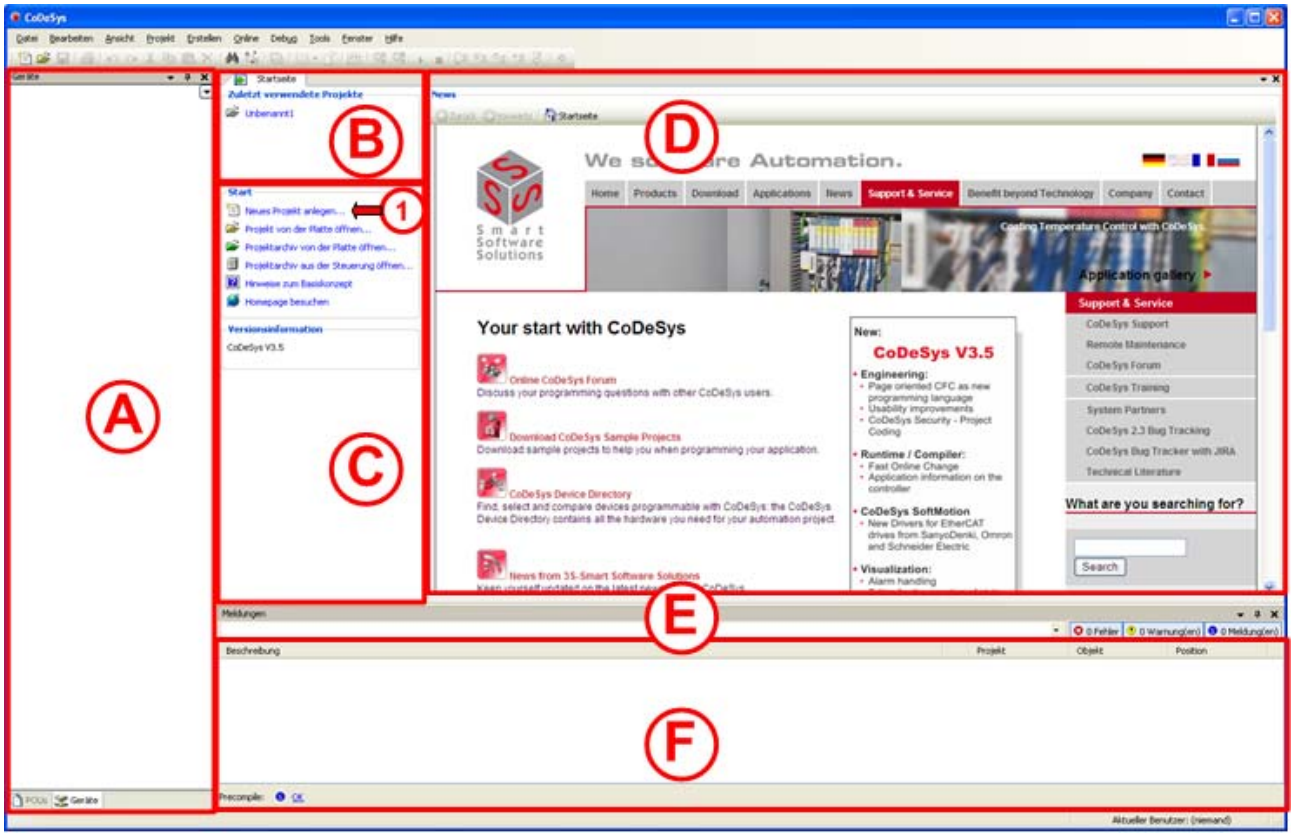



*Abbildung 1: CoDeSys Hauptfenster*

Das Hauptfenster besteht aus mehreren Teilfenstern:

Ⓐ Enthält eine Übersicht der im Projekt verwendeten Geräte.

Die Teilfenster Ⓑ, Ⓒ und Ⓓ bilden die **Startseite**. Diese besteht aus:

Ⓑ Bereich, in dem die zuletzt verwendeten Objekte aufgelistet werden

Ⓒ Links für den Projektstart

Ⓓ HTML Browser mit Links zur 3S CoDeSys Homepage

Ⓔ Meldungsbereich

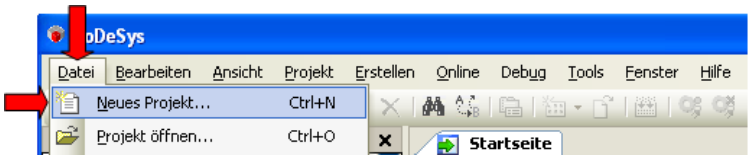
Ⓕ Beschreibungsbereich


© Hilscher, 2009 - 2014

## 4.3 Projekt anlegen, Gerät und Programmierart auswählen

1. Ein neues Projekt anlegen.

➢ Wählen Sie **Datei > Neues Projekt**




oder im Fensterbereich Ⓒ (Abbildung 1) den Eintrag **Neues Projekt anlegen** ①.

➢ Das Fenster **Neues Projekt** erscheint.

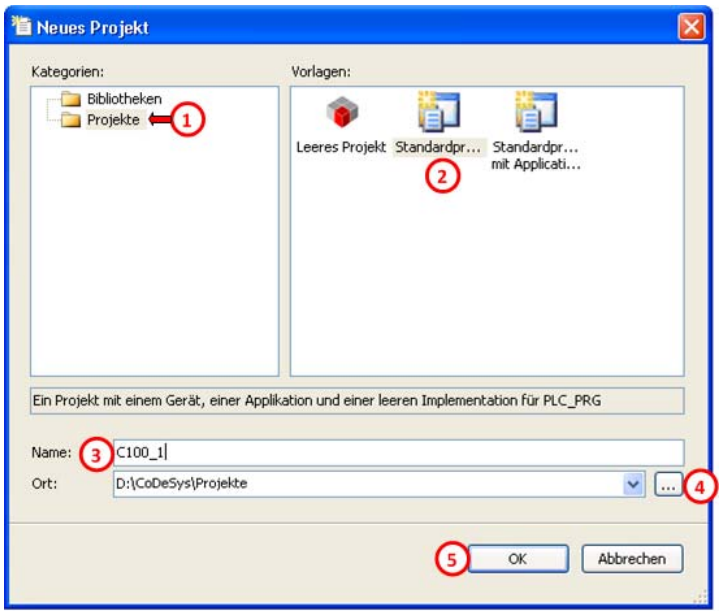



*Abbildung 2: Neues Projekt*

2. Angaben zum Projekt einstellen.

➢ Wählen Sie bei **Kategorien > Projekte** ①.

➢ Wählen Sie bei Vorlagen die **Vorlage > Standardprojekt** ②.

➢ Geben Sie im Feld **Name** ③ den Projektnamen z. B. ErstesProjekt ein.

➢ Stellen Sie im Feld **Ort** einen Ordner zur Ablage der Projektdateien z. B. D:\CoDeSys\Projekte ein. Verwenden Sie dazu die Browse-Schaltfläche ④.

➢ Klicken Sie **OK** ⑤.


© Hilscher, 2009 - 2014

Das Fenster **Neues Projekt** schließt sich und das Auswahlfenster **Standardprojekt** erscheint.

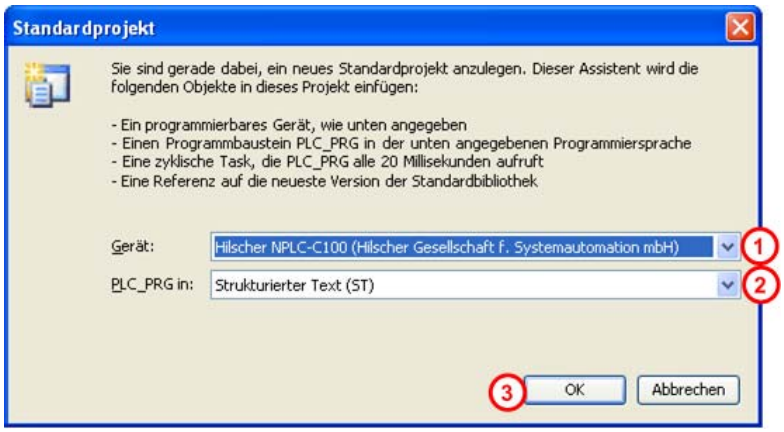

Standardprojekt
Sie sind gerade dabei, ein neues Standardprojekt anzulegen. Dieser Assistent wird die folgenden Objekte in dieses Projekt einfügen:
- Ein programmierbares Gerät, wie unten angegeben
- Einen Programmbaustein PLC_PRG in der unten angegebenen Programmiersprache
- Eine zyklische Task, die PLC_PRG alle 20 Millisekunden aufruft
- Eine Referenz auf die neueste Version der Standardbibliothek
Gerät:
Hilscher NPLC-C100 (Hilscher Gesellschaft f. Systemautomation mbH)
PLC_PRG in:
Strukturierter Text (ST)
1
2
3
OK
Abbrechen

*Abbildung 3: Standardprojekt, Geräteauswahl*

3. Angaben zum Standardprojekt einstellen.

➢ Wählen Sie im Popupfenster Standardprojekt (Abbildung 3) bei **Gerät** ① das Gerät netPLC Gerät aus, welches Sie in Ihrem Projekt einsetzen. Hier wurde das **NPLC-C100 (Hilscher GmbH)** gewählt.

➢ Wählen Sie bei **PLC_PRG in** die Programmierart **Strukturierter Text (ST)** ②.

➢ Klicken Sie **OK** ③.

Das Fenster **Standardprojekt** schließt sich und im Hauptfenster erscheint der Projektbaum im Fensterbereich Ⓐ.

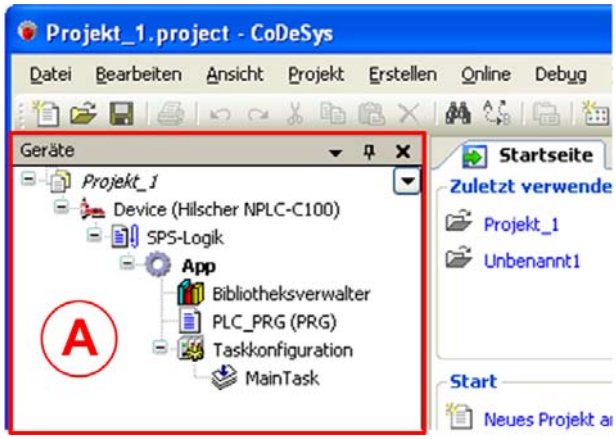



*Abbildung 4: Standardprojekt, Geräteübersicht*

4. Auf Geräteansicht umschalten.

➢ Prüfen Sie, ob die Ansicht Ⓐ auf Geräte steht (links unten im Anwendungsfenster).

➢ Falls die Ansicht auf **POUs** steht, schalten Sie auf die Geräteansicht um, indem Sie auf den Tab Geräte klicken.

Die Geräteansicht ist aktiviert.


© Hilscher, 2009 - 2014

5. Projekt speichern (zwischenspeichern).

➢ Wählen Sie **Datei > Speichern**, um das Projekt zu speichern.

➮ Das Projekt wird gespeichert.

### 4.3.1 Host E/As bei NPLC-C100 Slot-SPS verwenden

Bei der Slot-SPS NPLC-C100 haben Sie die Möglichkeit, vom Host aus einen separaten Ein- und Ausgangsbereich der SPS zu nutzen, der in den Speicherbereich des Host-PCs eingeblendet ist.
Ein Windows Anwendungsprogramm auf dem Host-PC beispielsweise kann direkt auf diese Ein- und Ausgangsdaten zugreifen und mit der SPS kommunizieren.

Die Eingangsdaten sind den Adressen %IB0 bis %IB511 und die Ausgangsdaten den Adressen %QB0 bis %QB511 zugeordnet, solange Slave-Geräte hinter der Host-E/A-Schnittstelle (Host_I_O_Interface) angehängt werden.

Damit Sie die Host-E/A-Schnittstelle im Steuerungsprogramm nutzen können, müssen Sie diese in das Projekt einfügen. Gehen Sie dazu wie folgt vor:

Host-Ein- und -Ausgänge anhängen.

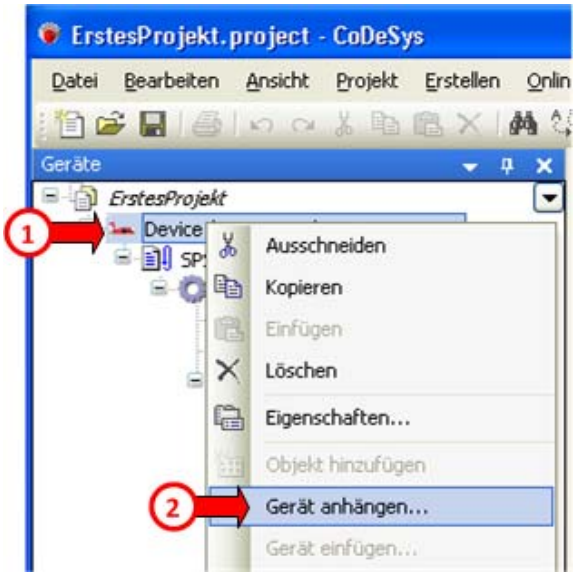



*Abbildung 5: NPLC-C100 Host-E/A anhängen (1)*

➢ Klicken Sie mit der rechten Maustaste auf **Device (NPLC-C100)** ①.

➢ Wählen Sie aus dem Kontextmenü **Gerät anhängen** ②.

➮ Das Popupfenster **Gerät anhängen** öffnet sich.


© Hilscher, 2009 - 2014

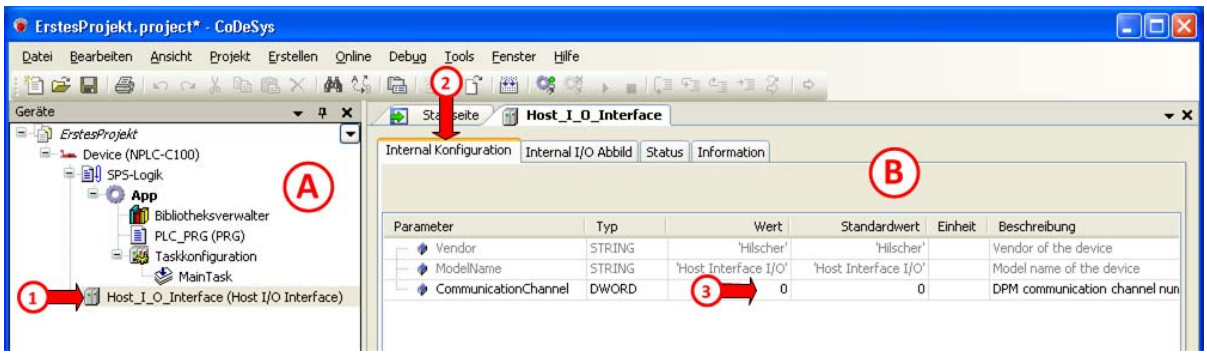



*Abbildung 6: NPLC-C100 Host-E/A anhängen (2)*

- Wählen Sie **Verschiedene** > **Host I/O Interface** (1).
- Klicken Sie **Gerät anhängen** (2).
- Klicken Sie **Schließen** (3).

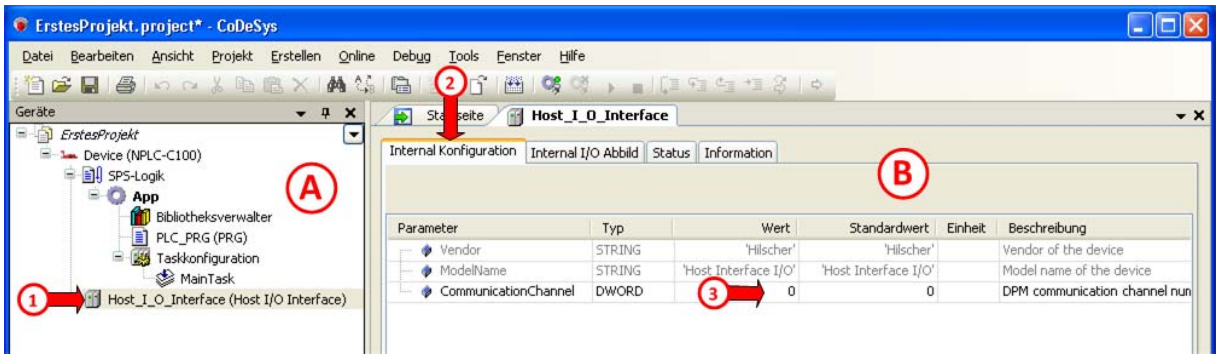



*Abbildung 7: C100 Host-E/As konfigurieren (1)*

- Im Gerätebaum (A) erscheint der Eintrag **Host_I_O_interface** (1).
- Öffnen Sie mit einem Doppelklick auf **Host_I_O_interface** (1) im Gerätebaum die Anzeige **Host_I_O_Interface** im Fensterbereich (B).
- Wählen Sie den Reiter **Internal Konfiguration** (2).
- Verwenden Sie für **CommunicationChannel** (3) den Defaultwert 0.


© Hilscher, 2009 - 2014

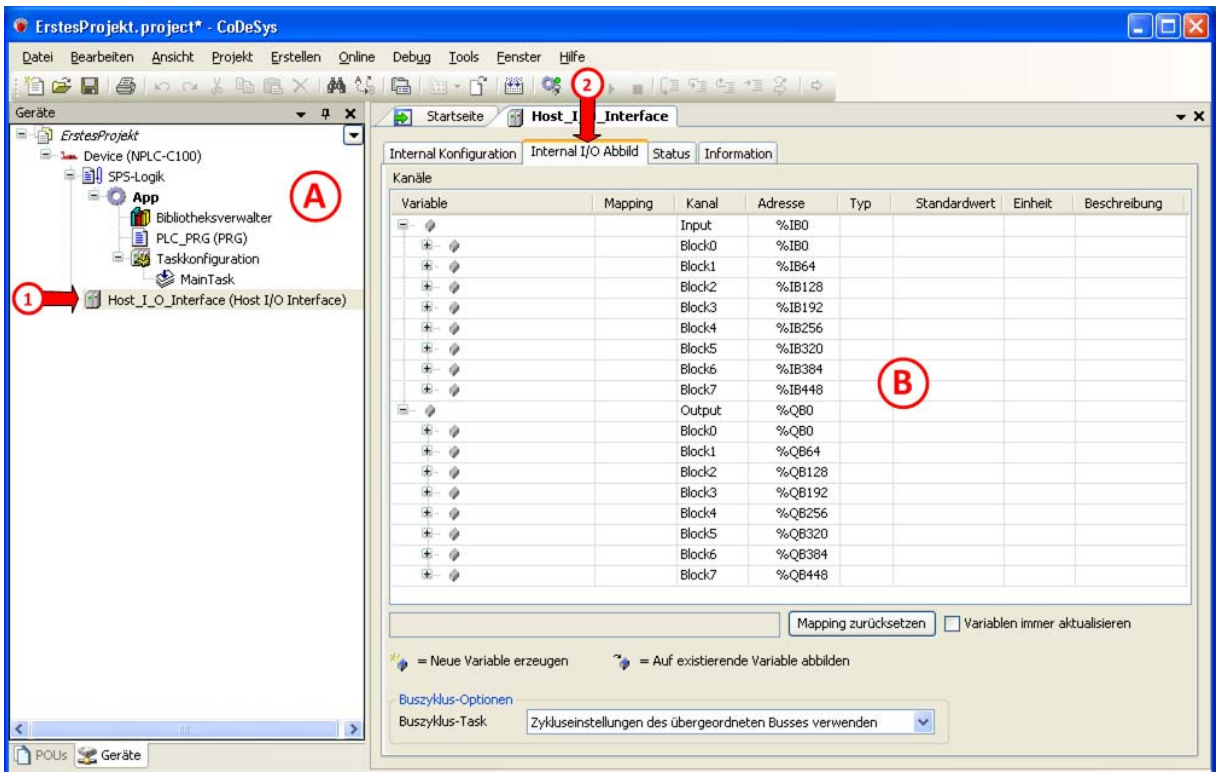



*Abbildung 8: C100 Host-E/As konfigurieren (2)*

- Falls noch nicht ausgewählt, öffnen Sie im Gerätebaum mit einem Doppelklick auf **Host_I_O_interface** (1) die Anzeige **Host_I_O_Interface** im Fensterbereich (B).
- Wählen Sie den Reiter **Internal I/O Abbild** (2).
- Nun können Sie mit einem Doppelklick in den einzelnen Tabellenfeldern die Signale bit- oder wortweise konfigurieren (Bereich (B)).

## 4.3.2 Lokale E/As beim NPLC-M100 Modul verwenden

Bei der Modul-SPS NPLC-M100 haben Sie die Möglichkeit, lokale Ein- und Ausgänge zu nutzen, die über die 100-Pin-Stiftleiste ausgeführt sind. Damit Sie diese im Steuerungsprogramm nutzen können, müssen Sie diese in das Projekt einfügen. Gehen Sie dazu wie folgt vor:

Lokale Ein- und Ausgänge anfügen.


© Hilscher, 2009 - 2014

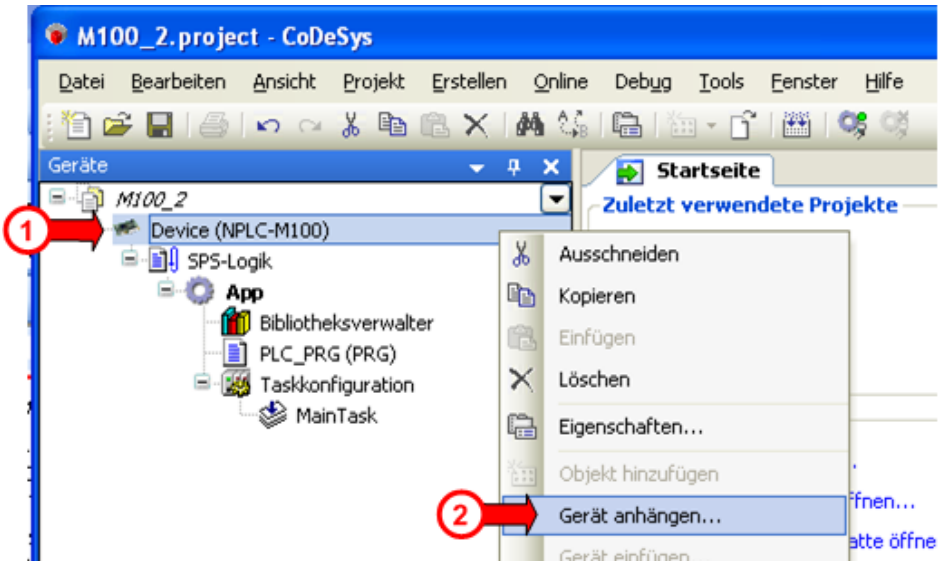



*Abbildung 9: M100 Lokale E/As anhängen (1)*

- Klicken Sie mit der rechten Maustaste auf **Device (NPLC-M100)** ①.
- Wählen Sie aus dem Kontextmenü **Gerät anhängen** ②.


© Hilscher, 2009 - 2014

Das Popupfenster **Gerät anhängen** öffnet sich.

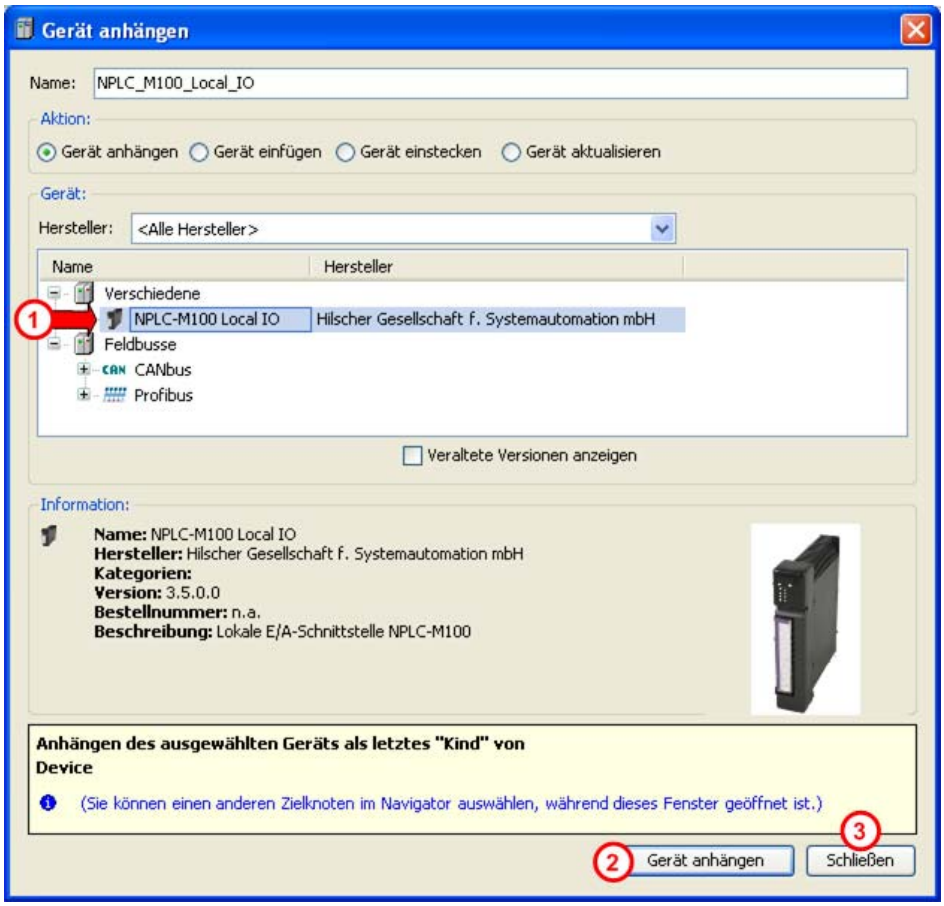



*Abbildung 10: M100 Lokale E/As anhängen (2)*

- Wählen Sie **Verschiedene > NPLC-M100 Local IO** ①.
- Klicken Sie **Gerät anhängen** ②.
- Klicken Sie **Schließen** ③.

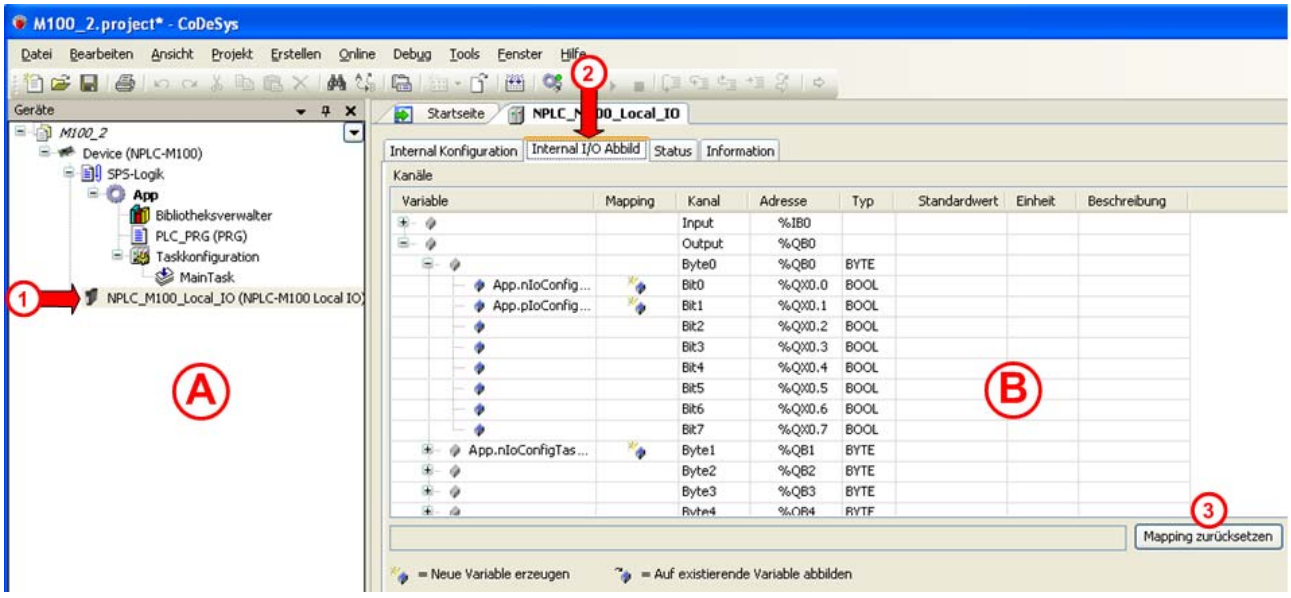



*Abbildung 11: NPLC-M100 Lokale E/As konfigurieren*


© Hilscher, 2009 - 2014

- Im Gerätebaum (A) erscheint der Eintrag **NPLC-M100 Local IO** (1).

➢ Öffnen Sie mit einem Doppelklick im Gerätebaum auf **NPLC-M100 Local IO** (1) die Signalkonfiguration im Fensterbereich (B).

➢ Wählen Sie den Reiter **Internal I/O Abbild** (2) aus.

- Nun können Sie mit einem Doppelklick in den einzelnen Tabellenfeldern die Signale bit- oder wortweise konfigurieren (Bereich (B)).


© Hilscher, 2009 - 2014

## 4.4 Buskonfiguration erstellen

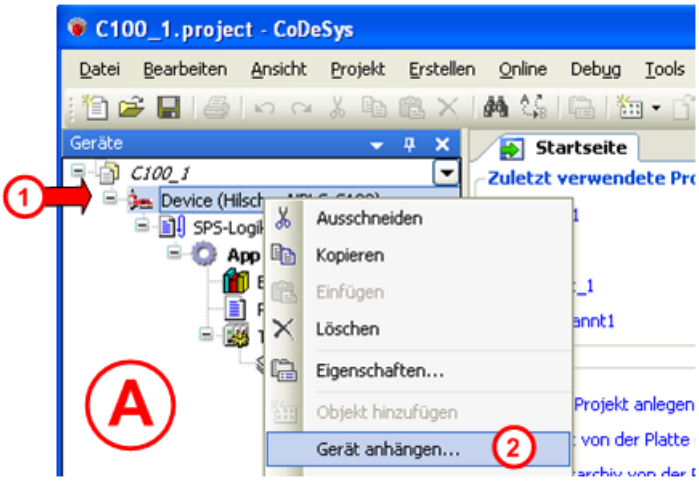
Für Ihr Feldbussystem ist eine Buskonfiguration erforderlich. Daher sind die folgenden Abschnitte *PROFIBUS DP Master konfigurieren*, *CANopen Master Konfigurationsfenster einfügen* und *DeviceNet Scanner konfigurieren* alternativ zu betrachten.

**Hinweis:** Zur Buskonfiguration erhalten Sie Unterstützung in der Online-Hilfe unter dem Hilfemenü: **Editoren > Geräte-Editoren > [Feldbusname]**.

### 4.4.1 Feldbussystem für die Steuerung auswählen

1. Feldbussystem für die Steuerung auswählen.

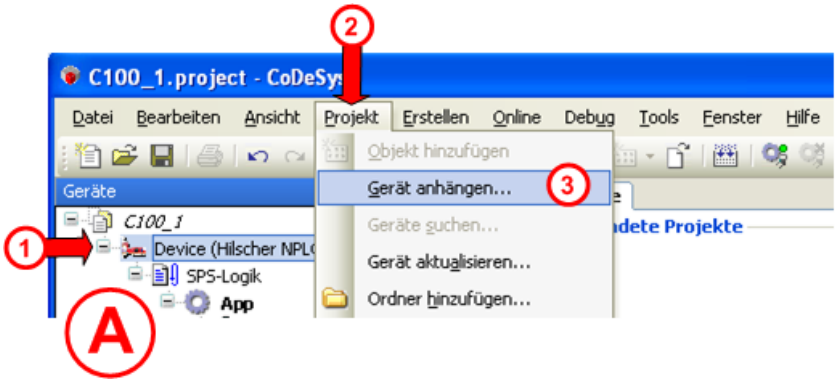
➢ Klicken Sie im Fensterbereich Ⓐ mit der rechten Maustaste auf die netPLC-Komponente ①.




➢ Wählen Sie im Kontextmenü **Gerät anhängen** ②.

oder:

➢ Klicken Sie im Fensterbereich Ⓐ mit der rechten Maustaste auf die netPLC-Komponente ①.




➢ Wählen Sie unter **Projekt** ② > **Gerät anhängen** ③.


© Hilscher, 2009 - 2014

Es öffnet sich folgendes Fenster:

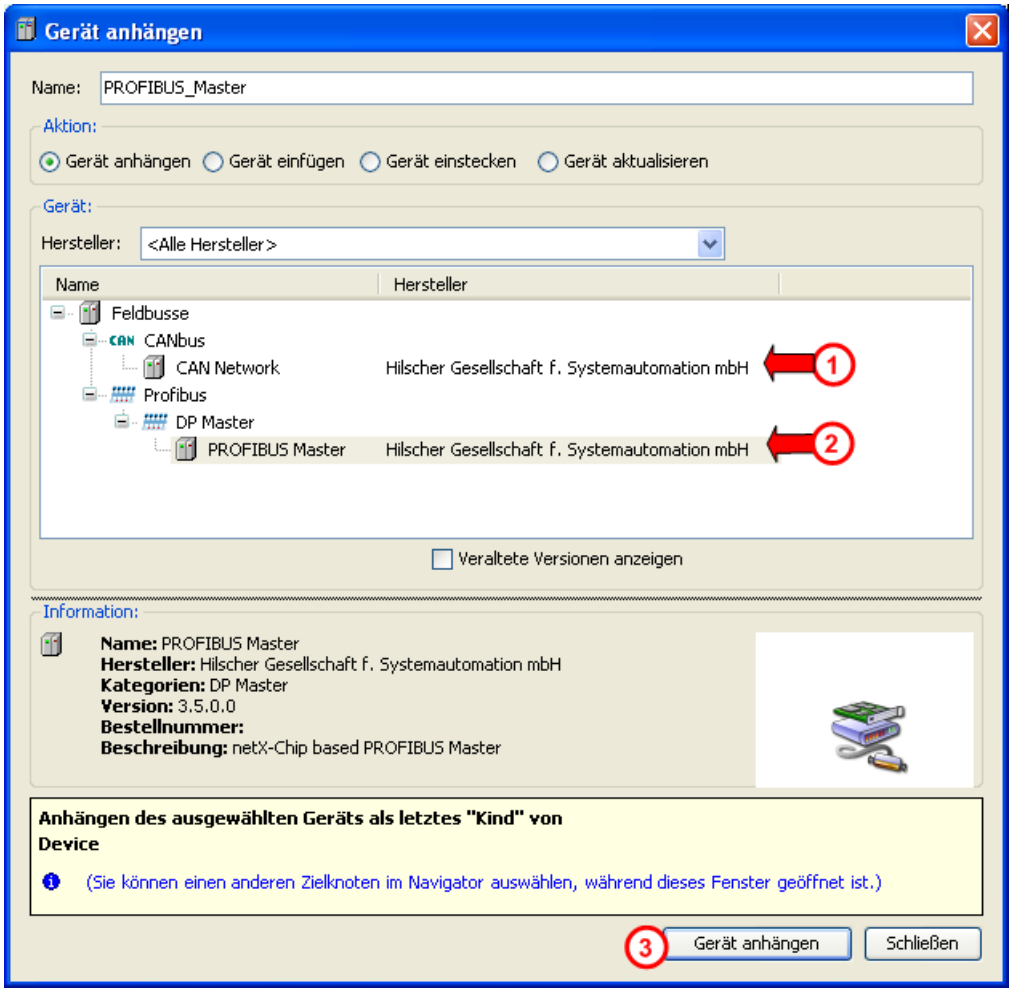


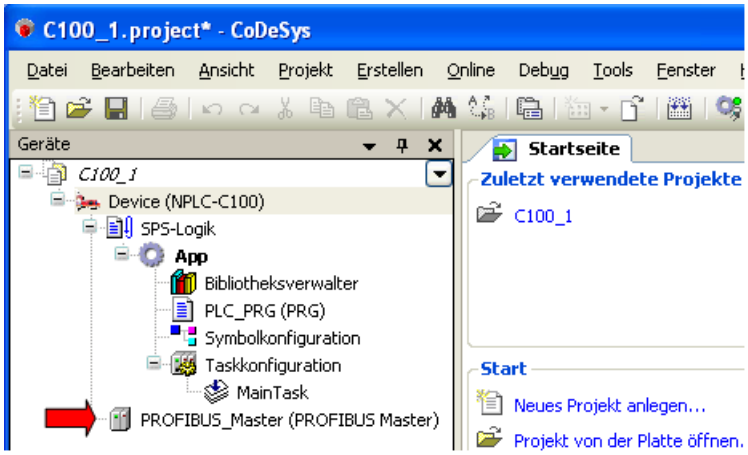

*Abbildung 12: Feldbus auswählen*

2. Feldbusauswahl

➢ Für CANopen oder DeviceNet wählen Sie **CAN Network** ①.
Für PROFIBUS wählen Sie **PROFIBUS Master** ②.

➢ Klicken Sie **Gerät anhängen** ③.

Der Gerätebaum hat sich um den Eintrag des Bussystems erweitert.




Für CANopen oder DeviceNet ist dieser Schritt zweistufig.


© Hilscher, 2009 - 2014

### 4.4.2 PROFIBUS DP Master konfigurieren

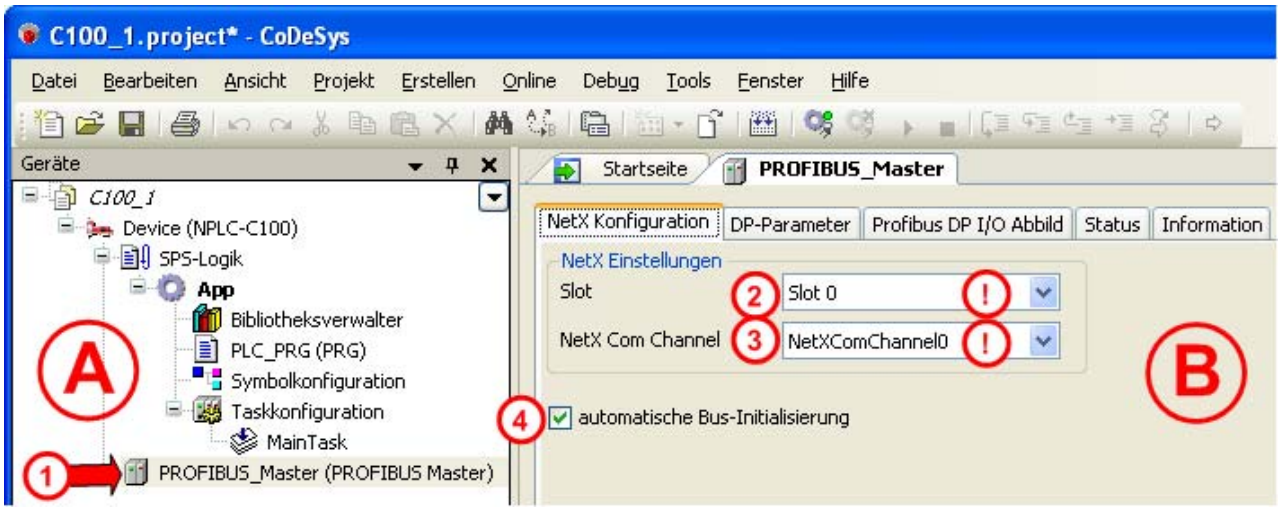



*Abbildung 13: PROFIBUS Master NetX Konfiguration*

- Öffnen Sie mit einen Doppelklick auf den **PROFIBUS Master** (1) im Gerätebaum (A) die Konfigurationsdialoge für die PROFIBUS-Parameter.

Es öffnet sich im Fensterbereich (B) der erste Reiter **NetX Konfiguration**.

Alle oben dargestellten Standardparameter müssen in der Default-Einstellung erhalten bleiben!

**Slot 0** (2), **NetXComChannel0** (3), der Haken bei **automatische Bus-Initialisierung** (4) muss erhalten bleiben.

Unter dem Reiter DP-Parameter finden Sie folgende Einstellmöglichkeiten:

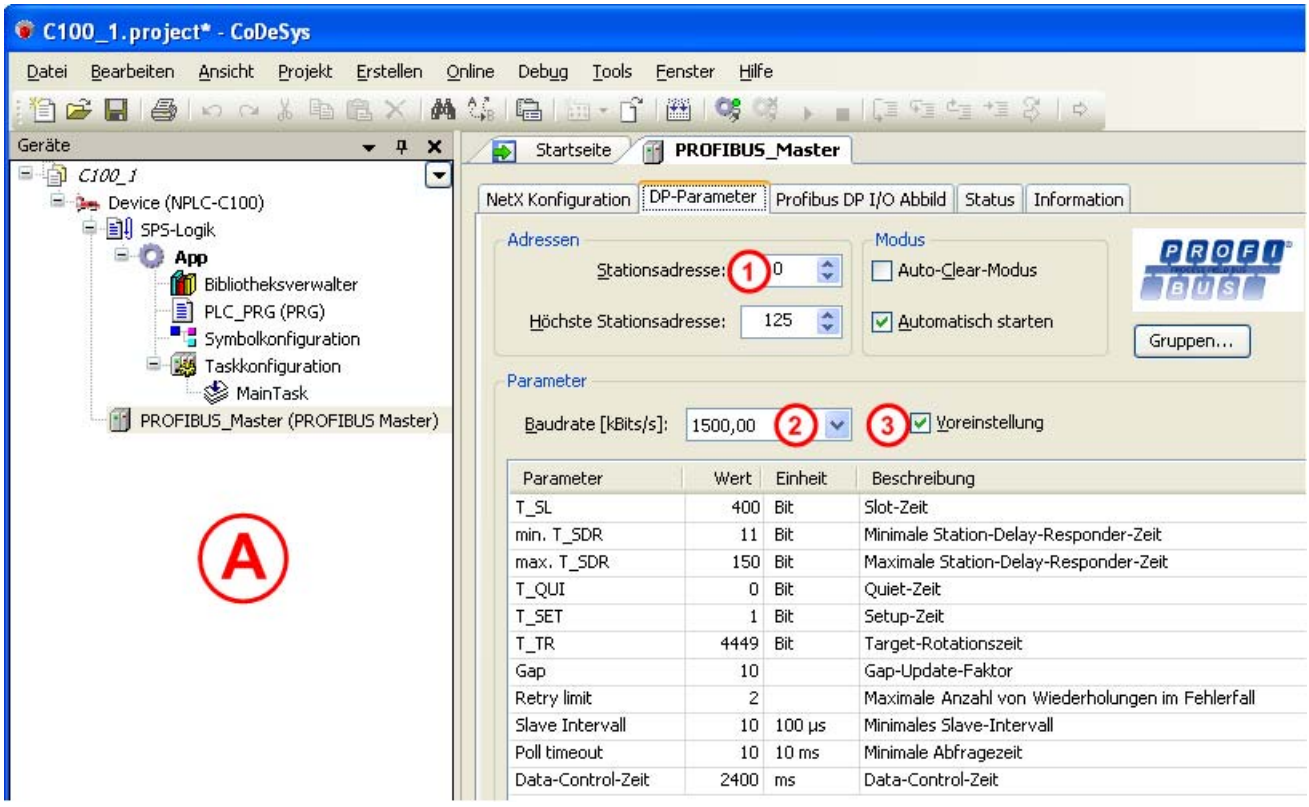



*Abbildung 14: PROFIBUS Master: DP-Parameter Konfiguration*


© Hilscher, 2009 - 2014

- Stellen Sie die **Stationsadresse** ① ein.
- Stellen Sie die **Baudrate** ② ein.
- Alle übrigen Parameter lassen sich nur einstellen, wenn der Haken bei **Voreinstellungen** ③ weggenommen wird.

### 4.4.3 CANopen Master Konfigurationsfenster einfügen

Die Konfiguration des CANopen Masters erfolgt in mehreren Stufen.

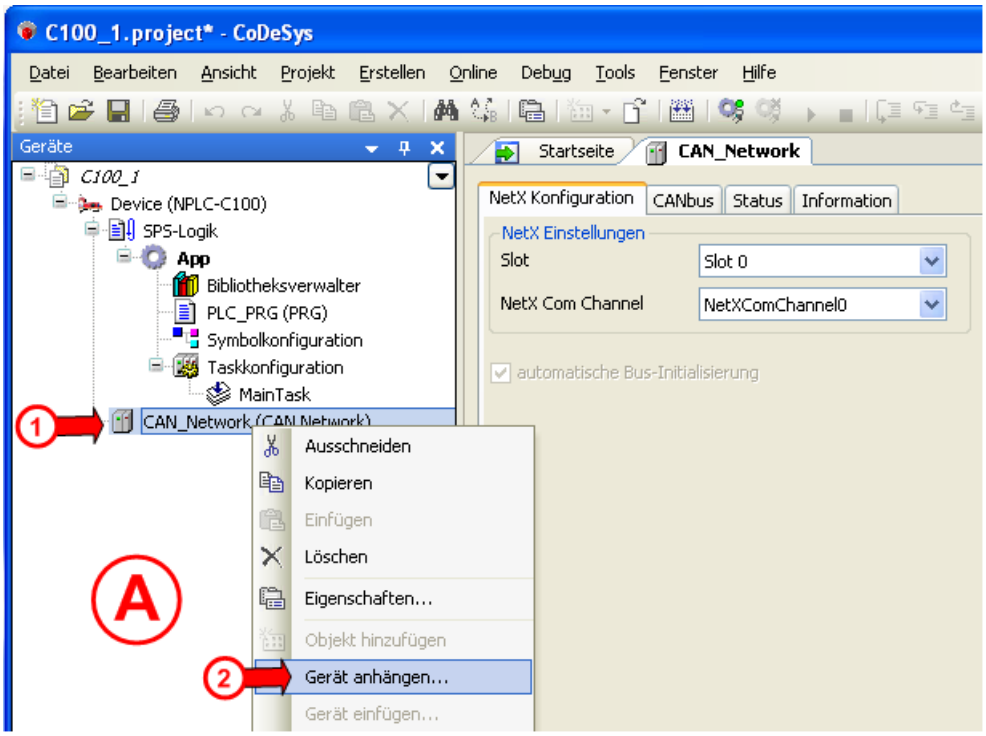



*Abbildung 15: CANopen 2. Dialog auswählen*

Nachdem im Abschnitt *Buskonfiguration erstellen* (Seite 25) der CANbus im Gerätebaum eingefügt wurde, muss nun

- im Gerätebaum Ⓐ mit der Maus der Eintrag **CAN_Network** ① mit der rechten Maustaste angeklickt werden. Im Kontextmenü den Eintrag **Gerät anhängen** ② wählen.


© Hilscher, 2009 - 2014

Es öffnet sich folgendes Dialogfenster:

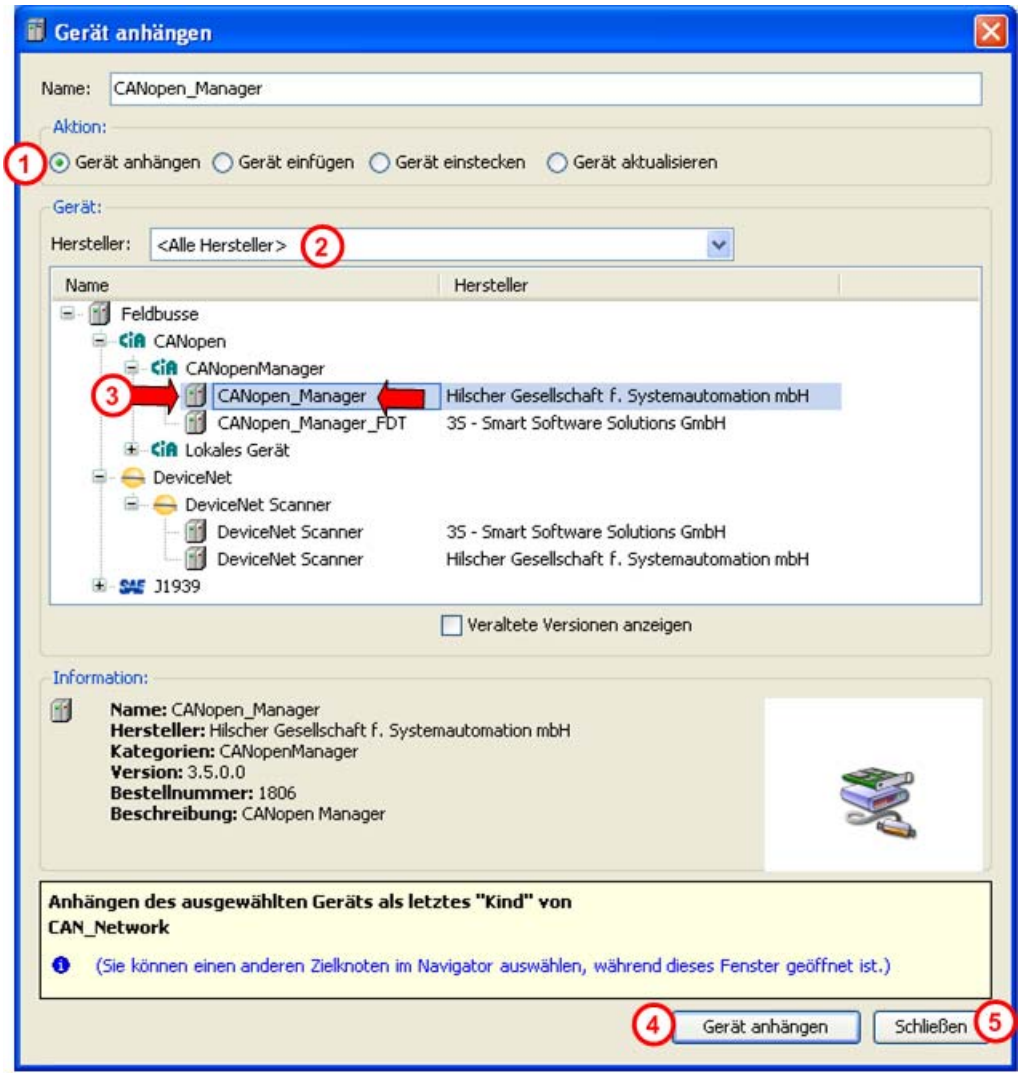



*Abbildung 16: CANopen Manager*

Wählen Sie folgende Einstellungen:

1 **Gerät anhängen**

2 **<Alle Hersteller>**

- Wählen Sie **CANopen_Manager** 3.
- Klicken Sie **Gerät anhängen** 4.
- Klicken Sie **Schließen** 5.

Das Dialogfenster schließt sich.

Die Dialogfenster für die Konfiguration des CANopen Busses sind zusammengesammelt.

Der Gerätebaum hat sich wie folgt erweitert.


© Hilscher, 2009 - 2014

#### 4.4.3.1 CANopen Master konfigurieren

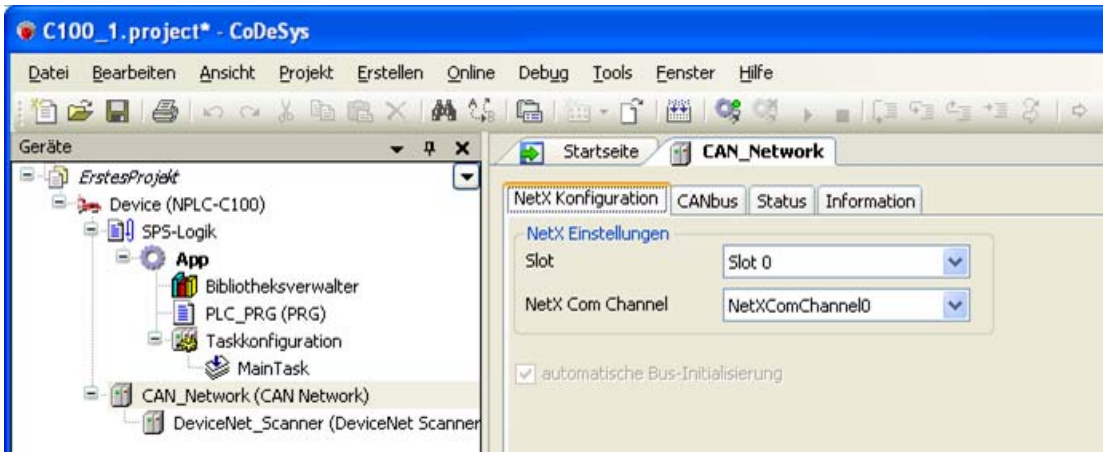



*Abbildung 17: NetX_CANbus > NetX Konfiguration*

Nehmen Sie unter **CAN_Network** ① > **NetX Konfiguration** ② folgende Einstellungen vor:

- **Slot** ③**:** Stellen Sie immer **Slot 0** ein.
- **NetX Com Channel** ④: Stellen Sie immer **NetXComChannel0** ein.

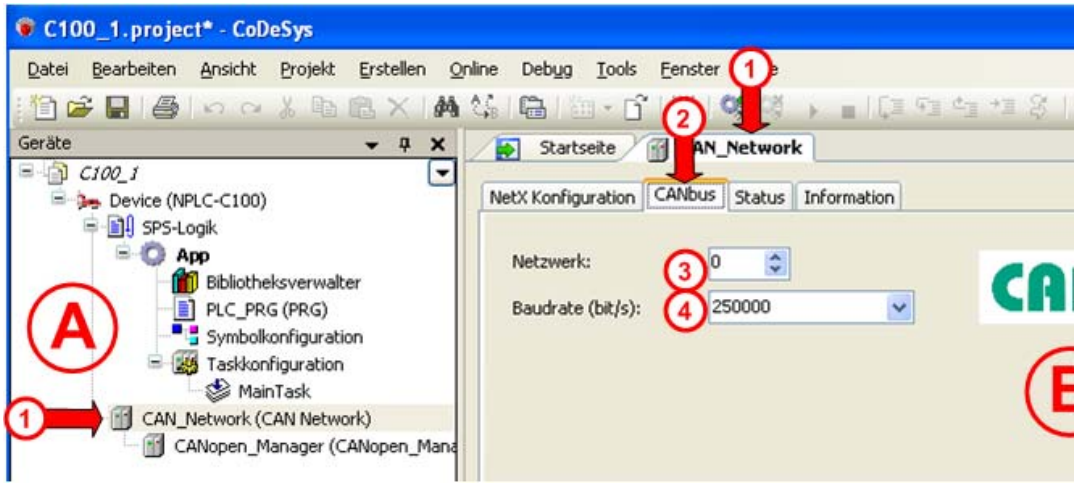



*Abbildung 18: NetX_CANbus > CANbus*

Nehmen Sie unter **CAN_Network** ① > **CANbus** ② folgende Einstellungen vor:

- **Netzwerk** ③**:** Stellen Sie immer **0** ein.
- **Baudrate** ④: Sie können folgende Baudrateneinstellungen wählen: 10000, 20000, 50000, 100000, 125000, 250000, 500000, 800000 und 1000000.


© Hilscher, 2009 - 2014

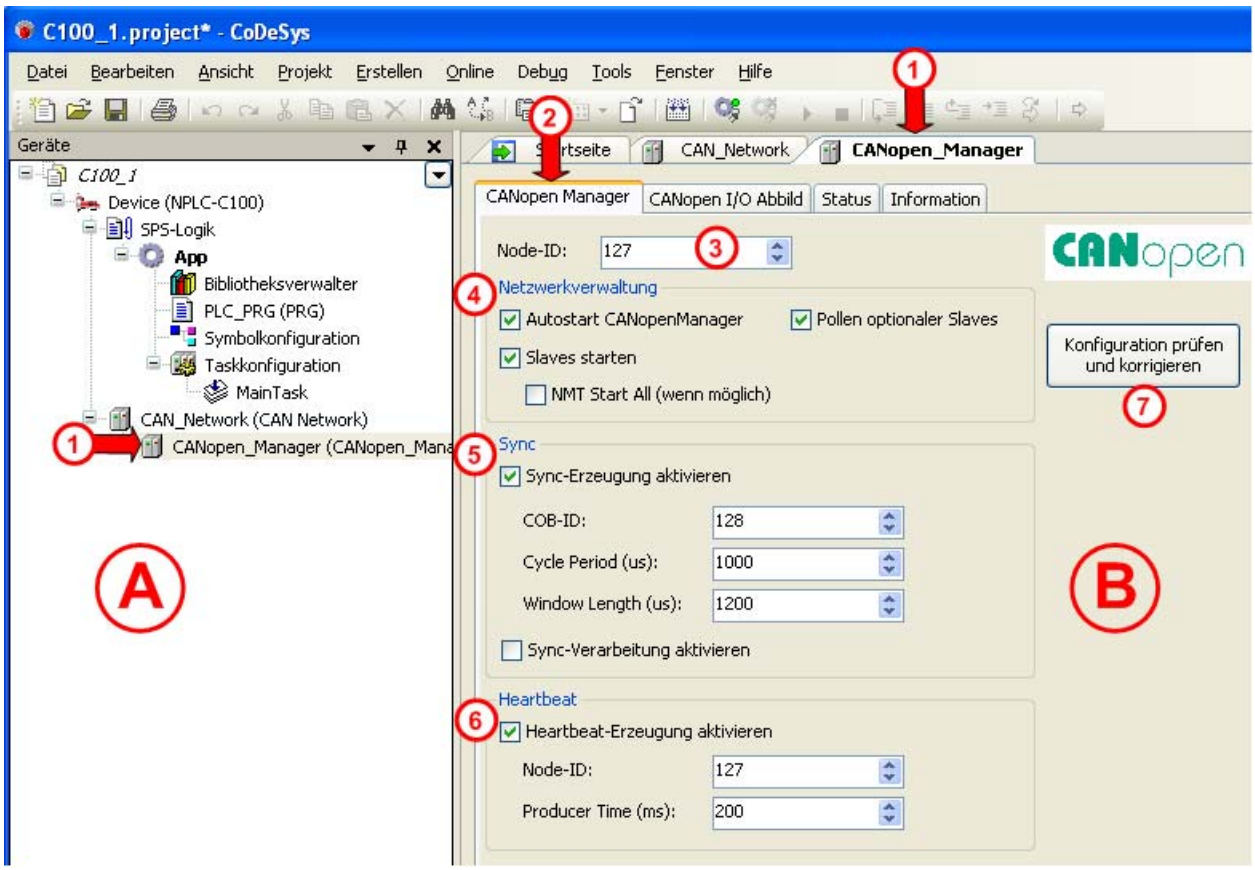



*Abbildung 19: CANopen_Manager > CANopen_Manager*

Um die Fenster erstmals zu öffnen, machen Sie im Fensterbereich Ⓐ einen Doppelklick auf **CANopen_Manager** ①.

Nehmen Sie unter **CANopen_Network** ① > **CANopen_Manager** ② folgende Einstellungen vor:

- **Node-ID** ③: 127
- Netzwerkverwaltungseinstellungen ④
- Synchronisationsverhalten ⑤
- Heartbeat ⑥

Nach erfolgter Einstellung können Sie die Plausibilität der gemachten Einstellungen durch einen Klick auf **Konfiguration prüfen und korrigieren** ⑦ testen.


© Hilscher, 2009 - 2014

### 4.4.4 DeviceNet Scanner konfigurieren

Die Konfiguration des DeviceNet Scanners erfolgt in mehreren Stufen.

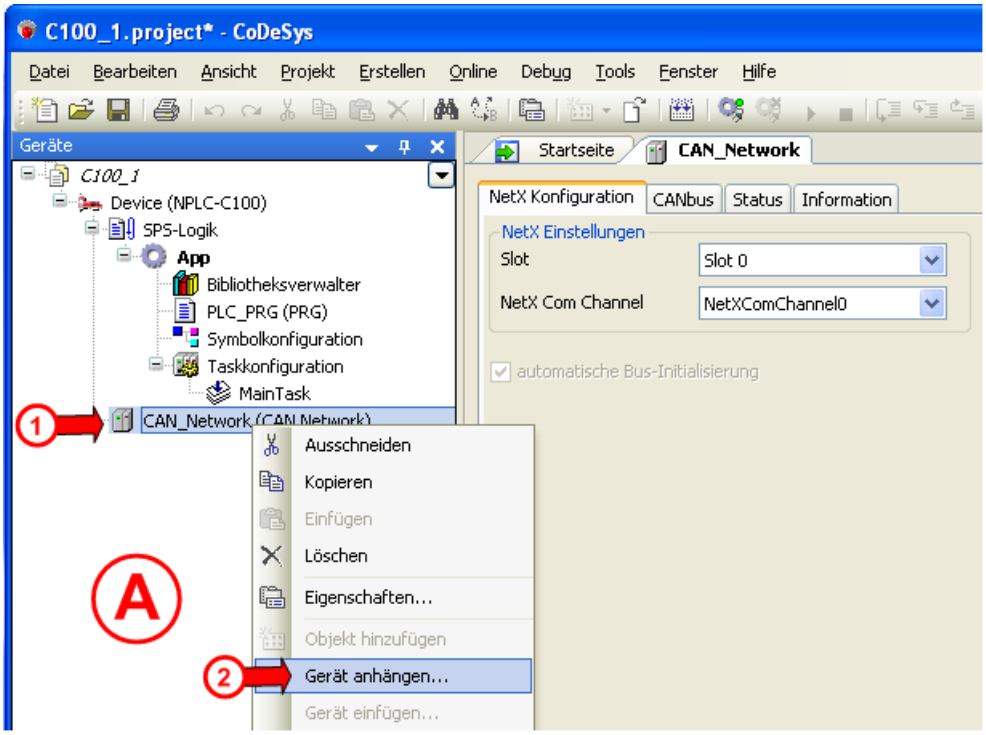



*Abbildung 20: CANopen > Gerät anhängen*

Nachdem im Abschnitt *Buskonfiguration erstellen* (Seite 25) der CANbus im Gerätebaum eingefügt wurde, muss nun

- im Gerätebaum (A) mit der Maus der Eintrag **CAN_Network** (1) mit der rechten Maustaste angeklickt werden. Wählen Sie im Kontextmenü **Gerät anhängen** (2).


© Hilscher, 2009 - 2014

Es öffnet sich folgendes Dialogfenster:

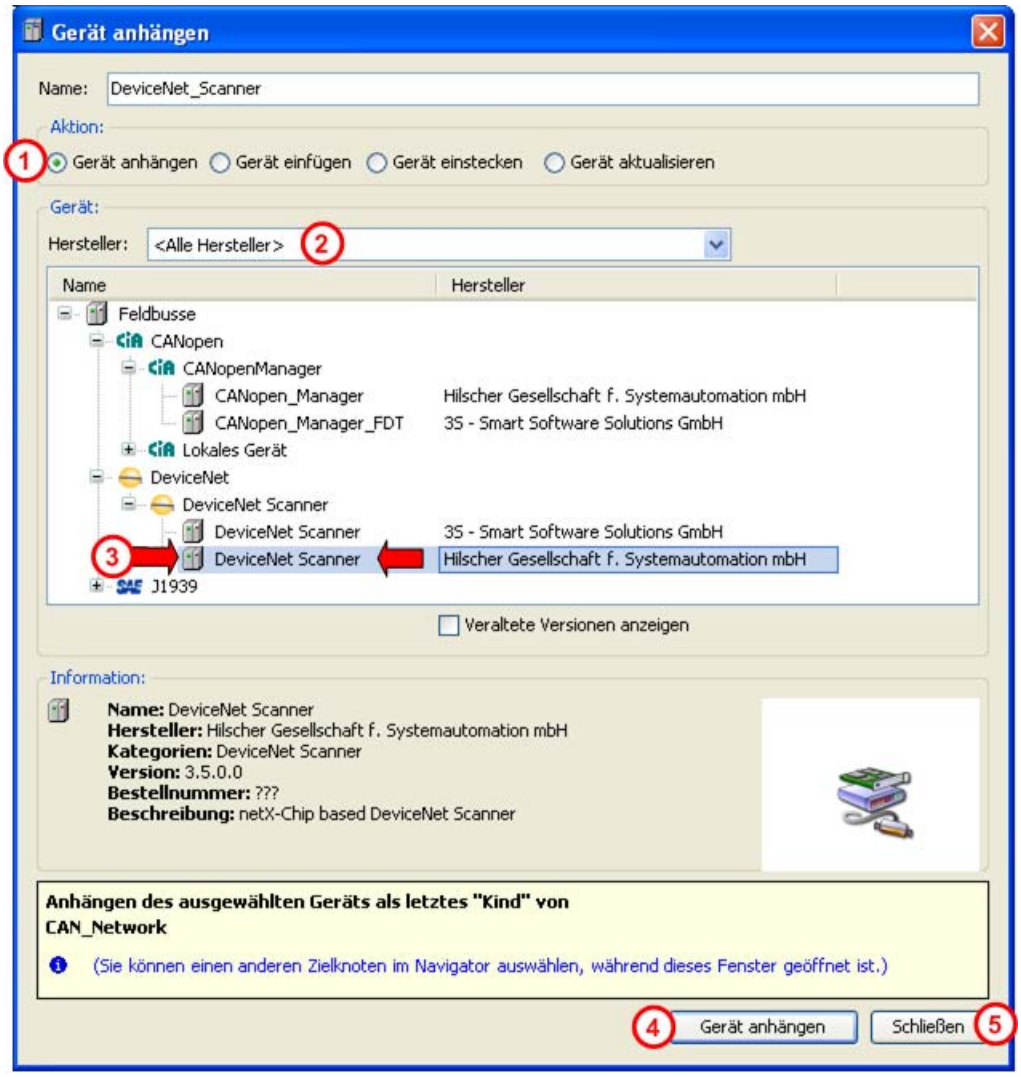



*Abbildung 21: CANopen Manager*

Wählen Sie folgende Einstellungen:

- Wählen Sie **Aktion** > **Gerät anhängen** ①.
- Wählen Sie **Gerät** > <**Alle Hersteller**> ②.
- Wählen Sie **DeviceNet Scanner** ③.
- Klicken Sie **Gerät anhängen** ④.
- Klicken Sie **Schließen** ⑤.

Das Dialogfenster schließt sich.

Die Dialogfenster für die Konfiguration des CAN Netzwerkes sind zusammengesammelt.


© Hilscher, 2009 - 2014

Der Gerätebaum hat sich wie folgt erweitert.

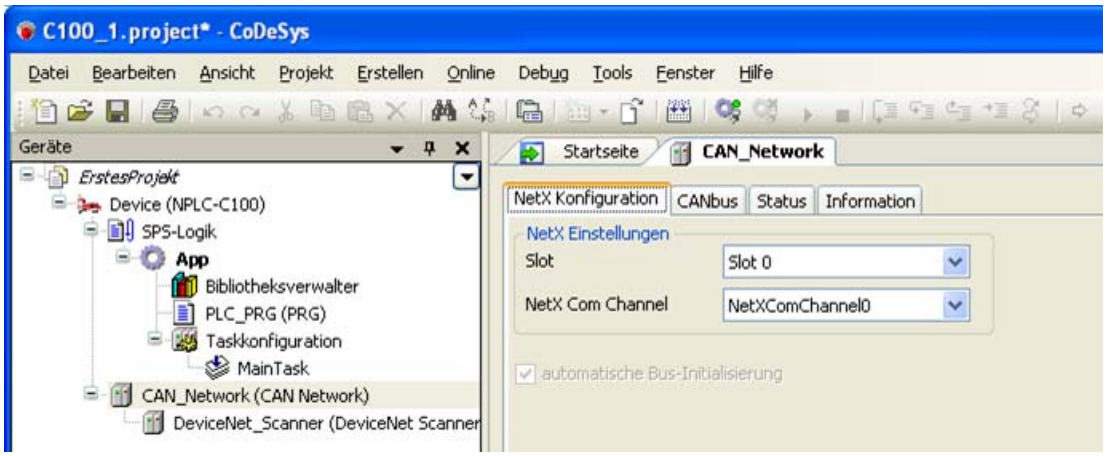



*Abbildung 22: NetX_CANbus > NetX Konfiguration*

Unter CAN_Network ① > NetX Konfiguration ② nehmen Sie folgende Einstellungen vor:

- **Slot** ③: Stellen Sie immer **Slot 0** ein.
- **NetX Com Channel** ④: Stellen Sie immer **NetXComChannel0** ein.

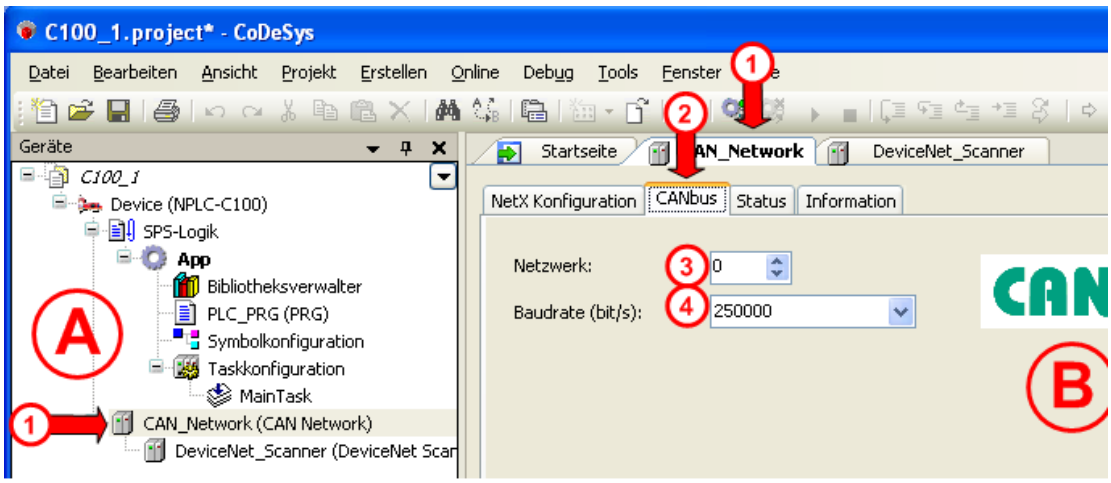



*Abbildung 23: NetX_CANbus > CANbus*

Nehmen Sie unter **CAN_Network** ① > **CANbus** ② folgende Einstellungen vor:

- **Netzwerk** ③: Stellen Sie immer **0** ein.
- **Baudrate** ④: Sie können folgende Baudrateneinstellungen wählen: 10000, 20000, 50000, 100000, 125000, 250000, 500000, 800000 und 1000000.


© Hilscher, 2009 - 2014

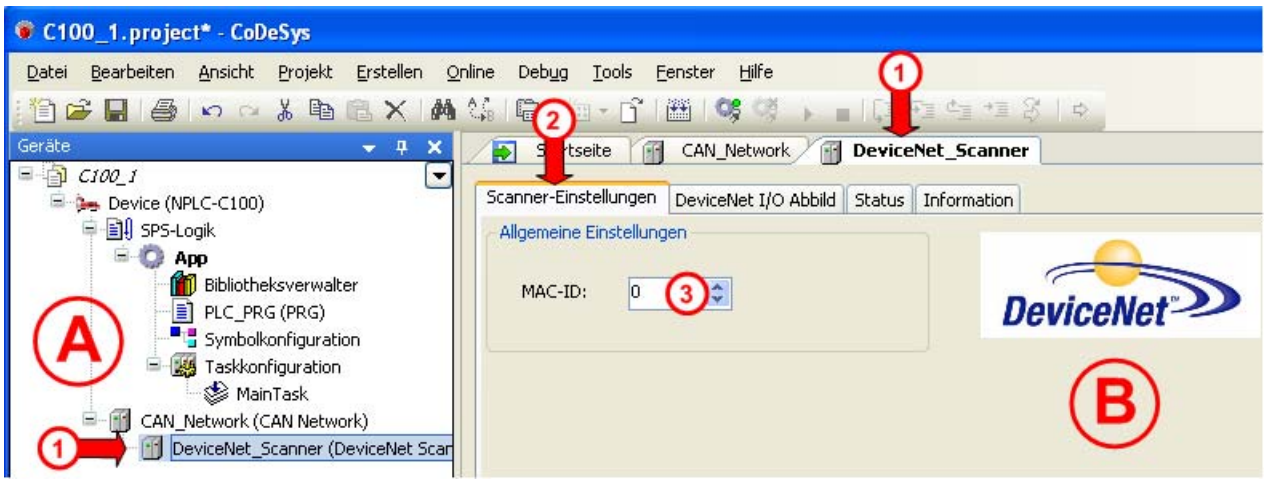



*Abbildung 24: Geräte-Feldbusadresse einstellen*

Um die Fenster erstmals zu öffnen, machen Sie im Fensterbereich Ⓐ einen Doppelklick auf **DeviceNet_Scanner** ①.

Nehmen Sie unter **DeviceNet_Scanner** ① > **Scanner-Einstellungen** ② folgende Einstellungen vor:

- **MAC-ID** ③: Stellen Sie die Adresse des Gerätes ein.


© Hilscher, 2009 - 2014

# 4.5 Busaufbau abbilden

## 4.5.1 Slave Gerätedateien importieren

Dieser Abschnitt beschreibt den Import eines Slave-Gerätes basierend auf einer Gerätebeschreibungsdatei.

Beispielhaft wird hier ein PROFIBUS-System mit einem Slave verwendet.

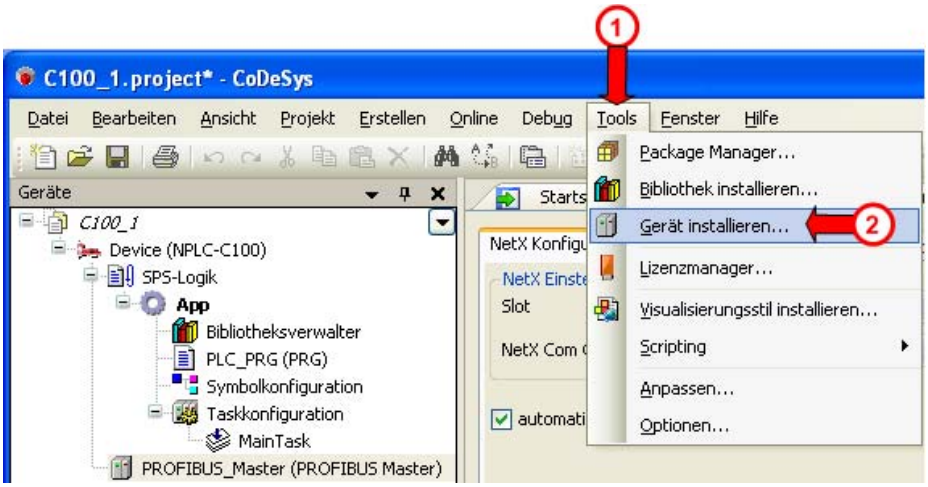



*Abbildung 25: Geräte-Beschreibungsdatei importieren*

- Wählen Sie unter **Tools** ① > **Gerät installieren** ②.
- Wählen Sie im sich öffnenden Dateimanager des Betriebssystems den Ort der Beschreibungsdatei aus.
- Wählen Sie **öffnen**.

Die Gerätebeschreibungsdatei wird importiert.

Nun kann dieses Gerät in der Buskonfiguration angehängt werden.


© Hilscher, 2009 - 2014

### 4.5.2 Slave-Gerät in den Busaufbau anhängen und konfigurieren

Das Anhängen eines Slave-Gerätes in die Buskonfiguration wird exemplarisch für PROFIBUS beschrieben. Der Ablauf ist für die anderen Bussysteme identisch.

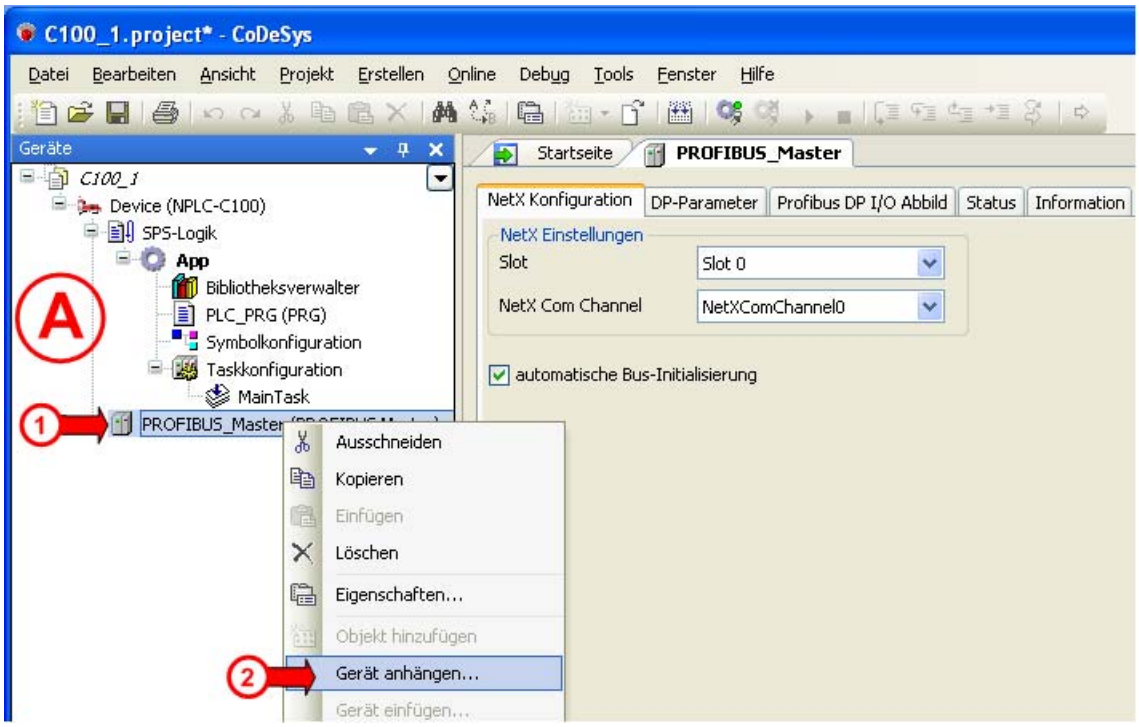



*Abbildung 26: Slave Gerät anhängen*

- Öffnen Sie im Gerätebaum (A) das Kontextmenü des Master-Gerätes (1).
- Wählen Sie **Gerät anhängen** (2).


© Hilscher, 2009 - 2014

➢ Es öffnet sich das Auswahlfenster **Gerät anhängen**.

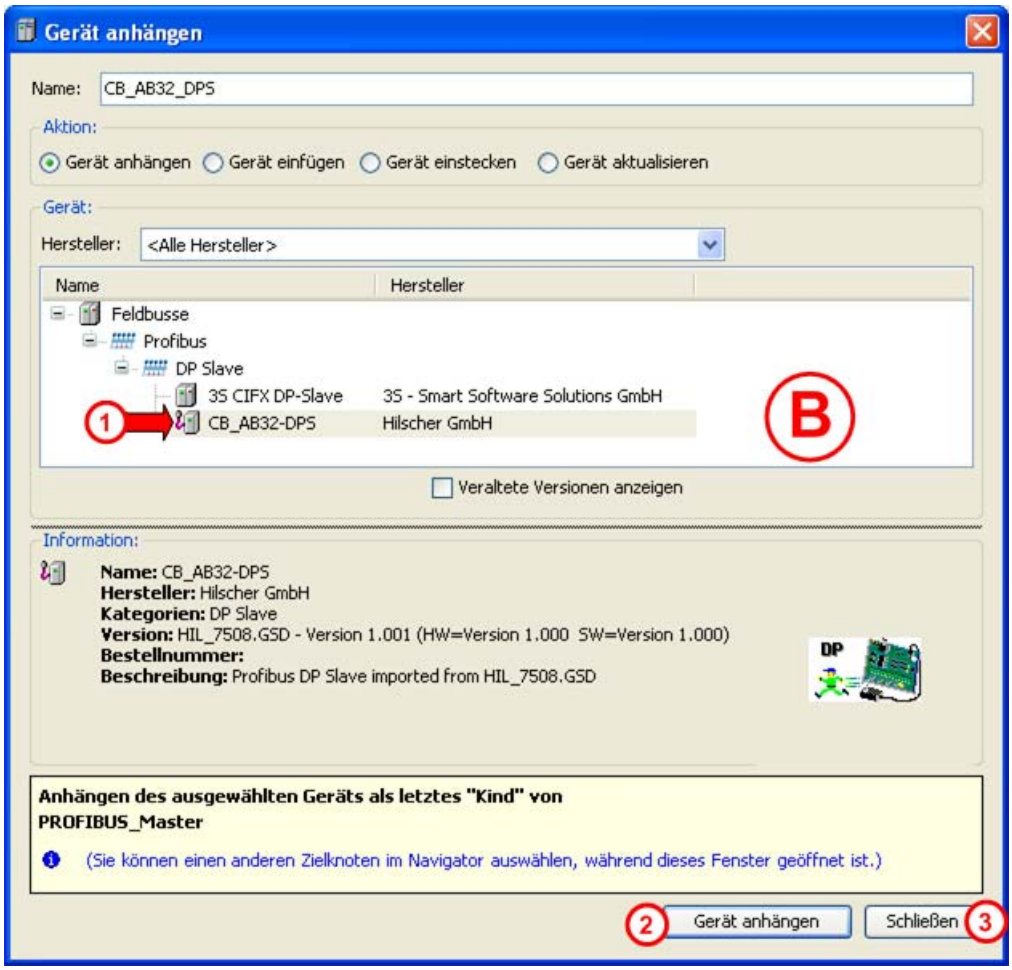



*Abbildung 27: Slave Gerät auswählen*

Im Fensterbereich Ⓑ der obigen Abbildung werden Ihnen alle verfügbaren Geräte des entsprechenden Feldbussystems aufgelistet.

➢ Wählen Sie Ihr Slave-Gerät ① aus.

➢ Klicken Sie **Gerät anhängen** ②.

➢ Klicken Sie **Schließen** ③.

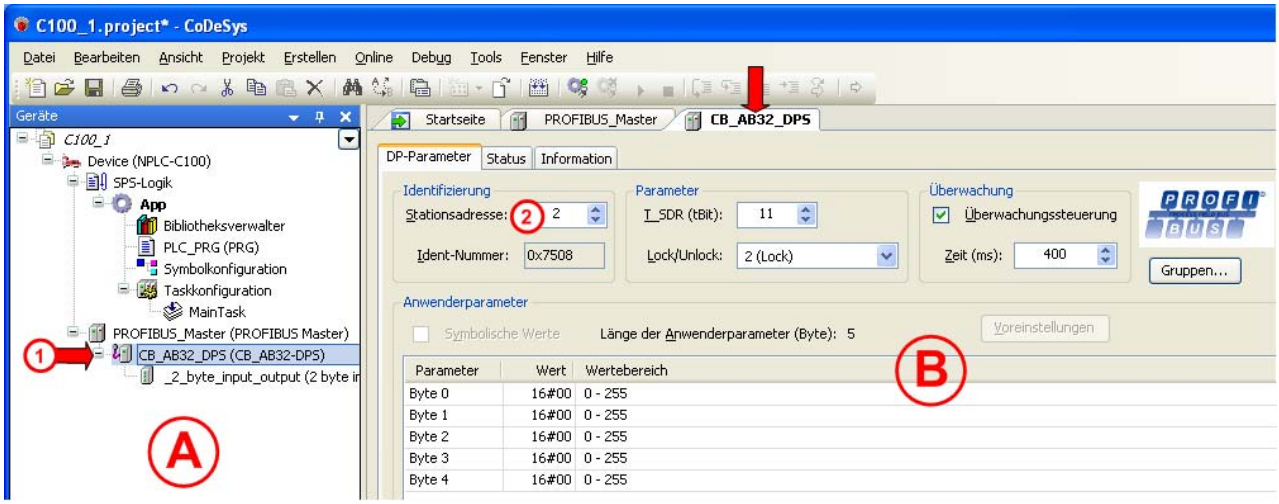



*Abbildung 28: Slave Gerät eingefügt*


© Hilscher, 2009 - 2014

Das Slave-Gerät wurde im Gerätebaum (Fensterbereich A) bei 1 eingefügt.

- Machen Sie einen Doppelklick auf den Eintrag 1.
- Es öffnet sich der Konfigurationsdialog für den Slave wie im Fensterbereich B dargestellt.
- Machen Sie hier alle erforderlichen Einstellungen.
- Stellen Sie die Stationsadresse für das Slave-Gerät 2 ein.

**Hinweis:** Sie machen hier nur die Einstellungen, die für den Master notwendig sind, um mit dem Slave kommunizieren zu können. Das Slave-Gerät selbst müssen Sie mit dem Konfigurationswerkzeug für das Slave-Gerät konfigurieren.

**Hinweis:** Im Kontextmenü, welches Sie durch Anwahl des **Gerätes** im Gerätebaum im Fensterbereich A aufrufen können, gibt es den Eintrag **Eigenschaften**. Hier können Sie dem Gerät im Gerätebaum einen sprechenden Namen geben.

## 4.5.3 Adressen Ein-/Ausgänge des Slaves ermitteln

Zur Ermittlung der E/A-Adressen des Slaves für das SPS-Programm gehen Sie wie folgt vor:

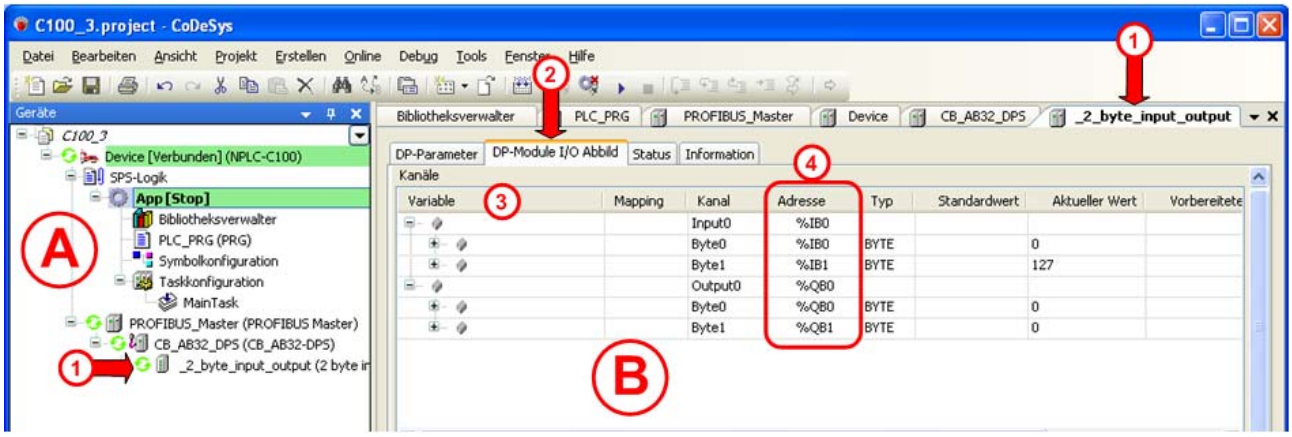



*Abbildung 29: Slave E/A-Adressen ermitteln*

- Öffnen Sie im Gerätebaum A mit einen Doppelklick auf den Eintrag der E/As 1 des Slaves die Anzeige.
- Es öffnet sich der Dialog der Slave E/As im Fensterbereich B der obigen Abbildung.
- Klicken Sie auf den Reiter **I/O Abbild** 2.
- Öffnen Sie, soweit erforderlich, die Signale in der Spalte **Variable** 3.
- In der Tabelle können Sie der Spalte **Adresse** 4 die Adressen entnehmen, mit denen die E/As im SPS-Programm angesprochen werden können.

**Hinweis:** Die Adressen 4 sind %IB0, %IB1, %QB0 und %QB1, wenn keine Host-E/A-Schnittstelle verwendet wird. Die Adressen 4 sind %IB512, %IB513, %QB512 und %QB513, wenn eine Host-E/A-Schnittstelle oberhalb des Master-Gerätes verwendet wird.


© Hilscher, 2009 - 2014

## 4.6 Steuerungsprogramm erstellen

1. Editor für Baustein PLC_PRG öffnen.

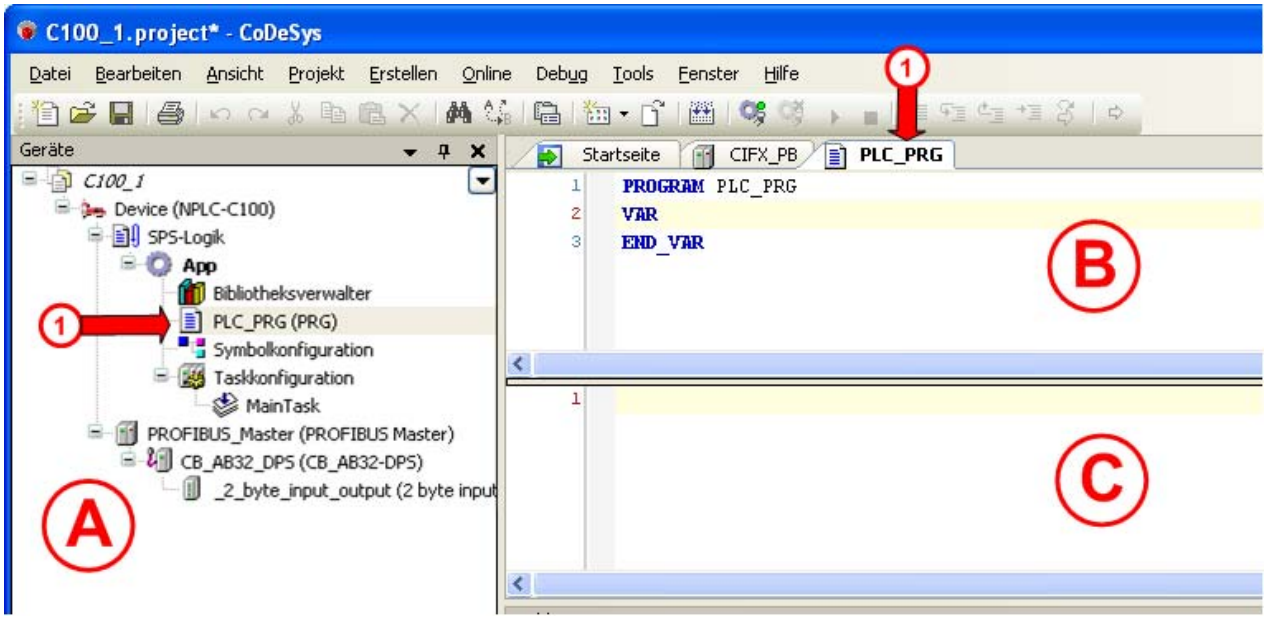



*Abbildung 30: Slave Gerät eingefügt*

➢ Öffnen Sie mit einem Doppelklick auf **PLC_PRG (PRG)** (① im Fensterbereich Ⓐ) den Editor.

**Hinweis:** Sie können im Programm-Editor nur editieren, wenn Sie nicht auf der netPLC-Komponente eingeloggt sind.

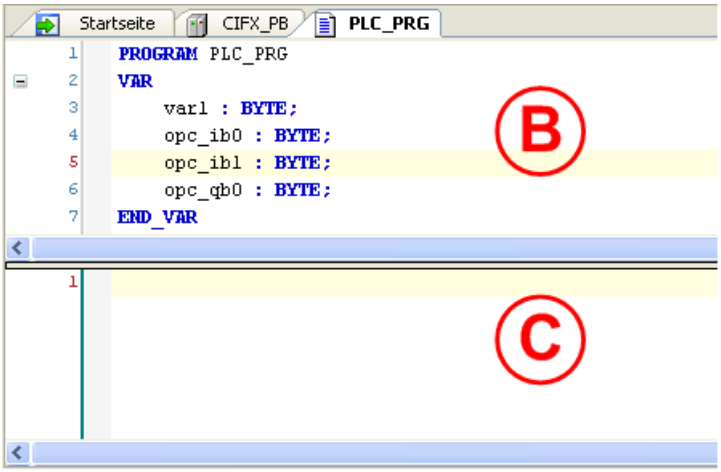
↬ Das Editorfenster erscheint. Das obige Bild zeigt den Editor noch ohne Variable Ⓑ und ohne SPS-Programm Ⓒ.

2. Variable anlegen

➢ Legen Sie folgende Variablen an:

```
var1 : BYTE;
opc_ib0 : BYTE;
opc_ib1 : BYTE;
opc_qb0 : BYTE;
```

↬



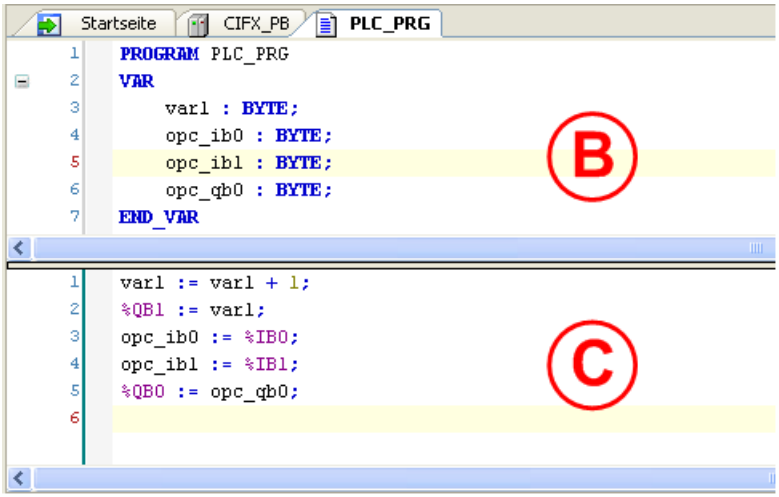


© Hilscher, 2009 - 2014

3. SPS-Programm editieren.

➢ Fügen Sie folgendes SPS-Programm ein:

```
var1 := var1 + 1;
%QB1 := var1;
opc_ib0 := %IB0;
opc_ib1 := %IB1;
%QB0 := opc_qb0;
```




Kurze Erläuterung zum Programmbeispiel:

`var1 := var1 + 1;`

Die Variable var1 wird mit jedem SPS Zyklus um 1 erhöht.

`%QB1 := var1;`

Der Wert von Variable var1 wird auf das Ausgangsbyte %QB1 ausgegeben.

`opc_ib0 := %IB0;` und `opc_ib1 := %IB1;`

Eingangsbyte %IB0 und %IB1 werden auf die Variablen opc_ib0 bzw. opc_ib1 kopiert. Dies ist die Vorbereitung, dass der OPC-Server auf die Eingangsdaten zugreifen kann.

`%QB0 := opcqb0;`

Die Variablen opc_qb0 wird auf das Ausgangsbyte %QB0 kopiert. Dies ist die Vorbereitung, dass der OPC-Server auf die Ausgangsdaten zugreifen kann.

Anmerkung: Das Ausgangsbyte %QB1 wird von Steuerungsprogramm gesetzt und deshalb dem OPC-Server kein Zugriff gegeben.

**Hinweis:** Die Adressen %IB0 und %QB0 (Bereich Ⓒ) starten bei 0, wenn keine Host-E/A-Schnittstelle verwendet wird. Wenn eine Host-E/A-Schnittstelle oberhalb des Master-Gerätes verwendet wird, dann starten die Adressen ab 512, d. h. die Adressen lauten dann %IB512 und %QB512 (und höher).

4. Projekt speichern (zwischenspeichern).

➢ Speichern Sie das Projekt mit **Datei > Projekt speichern**.

Das Projekt wird gespeichert.


© Hilscher, 2009 - 2014

**Hinweis:** Zum Definieren remanenter Variablen:

Remanente Variable anlegen.

- Legen Sie den Bereich an:
  VAR RETAIN

  END_VAR
- Legen Sie innerhalb dieses Bereiches Variablen an:
  VAR RETAIN
  uiRemCount : UINT;
  END_VAR

```
1  PROGRAM PLC_PRG
2  VAR
3      var1 : BYTE;
4      opc_ib0 : BYTE;
5      opc_ib1 : BYTE;
6      opc_qb0 : BYTE;
7  END_VAR
8  VAR RETAIN
9      uiRemCount : UINT;
10 END_VAR
```

**Hinweis:** Persistente Variablen lassen sich über das Kontextmenü des **APP** Eintrages im Gerätebaum erstellen.

Weitere Informationen zu remanenten und persistenten Variablen finden Sie in der Hilfe zum CoDeSys-Programm.


© Hilscher, 2009 - 2014

### 4.6.1 Symbolkonfiguration erstellen

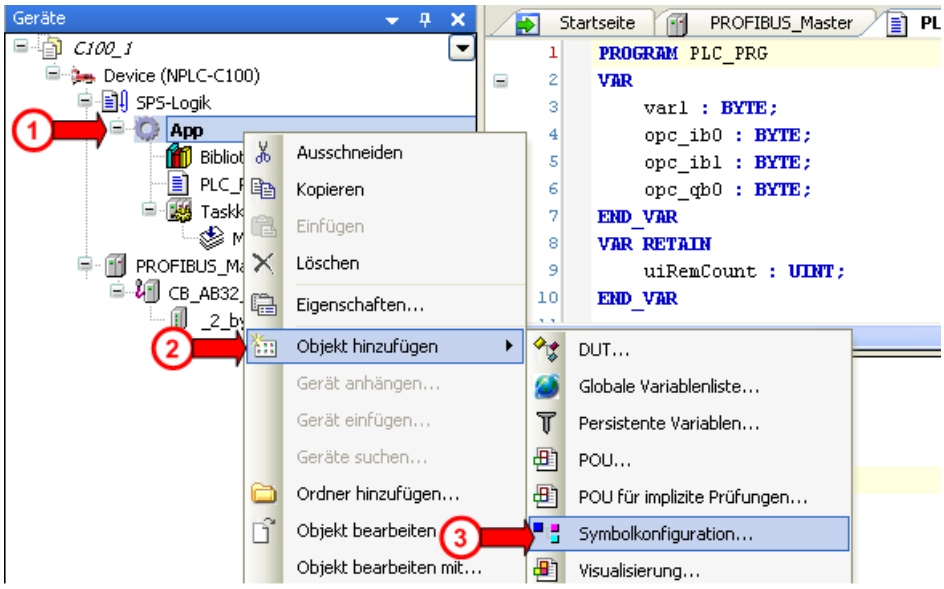
Die Symbolkonfiguration wird verwendet, um Symbole mit bestimmten Zugriffsrechten zu erzeugen, über die von extern auf Projekt(applikations)variablen zugegriffen werden kann, z.B. von einem OPC-Klient und/oder einer Visualisierung.

Die Symbolkonfiguration wird nach dem Build-Prozess beim Download des Projekts in das Zielgerät automatisch in eine Symbol-Datei im XML-Format geschrieben und im Projektverzeichnis abgespeichert. Sie erkennen diese Symbol-Datei an der Namenskonvention:
`<project name>.<device name>.<application name>.xml`

Diese Datei können Sie später in eine Applikation – wie z. B. die QViS-Visualisierungssoftware – importieren, damit diese auf die Variablen zugreifen kann.

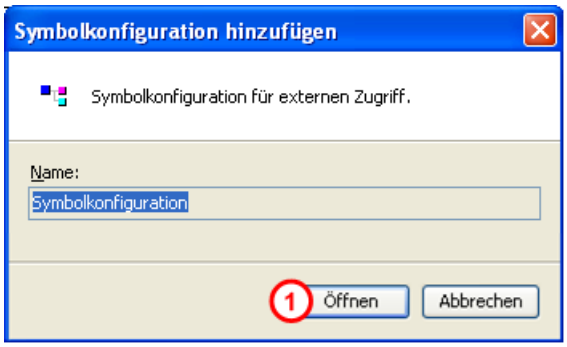
1. Objekt hinzufügen.

➢ Wählen Sie aus dem Kontextmenü des Gerätebaums **App,** dann den Eintrag **Objekt hinzufügen** und dann **Symbolkonfiguration**.




2. Objekt Symbolkonfiguration hinzufügen.

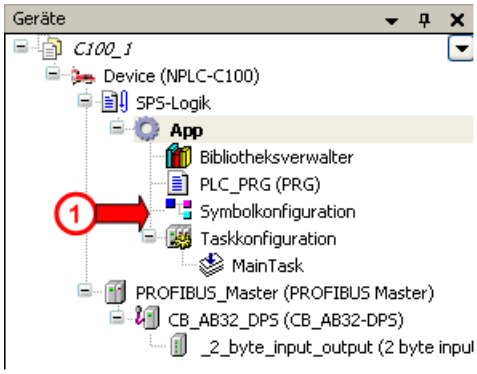
➢ Es öffnet sich folgendes Dialogfenster:




➢ Klicken Sie **Öffnen** ①.


© Hilscher, 2009 - 2014

➢ Im Projektbaum erscheint nun der Eintrag **Symbolkonfiguration**.




➢ Im Editorbereich erscheint der Eintrag **Symbolkonfiguration**.

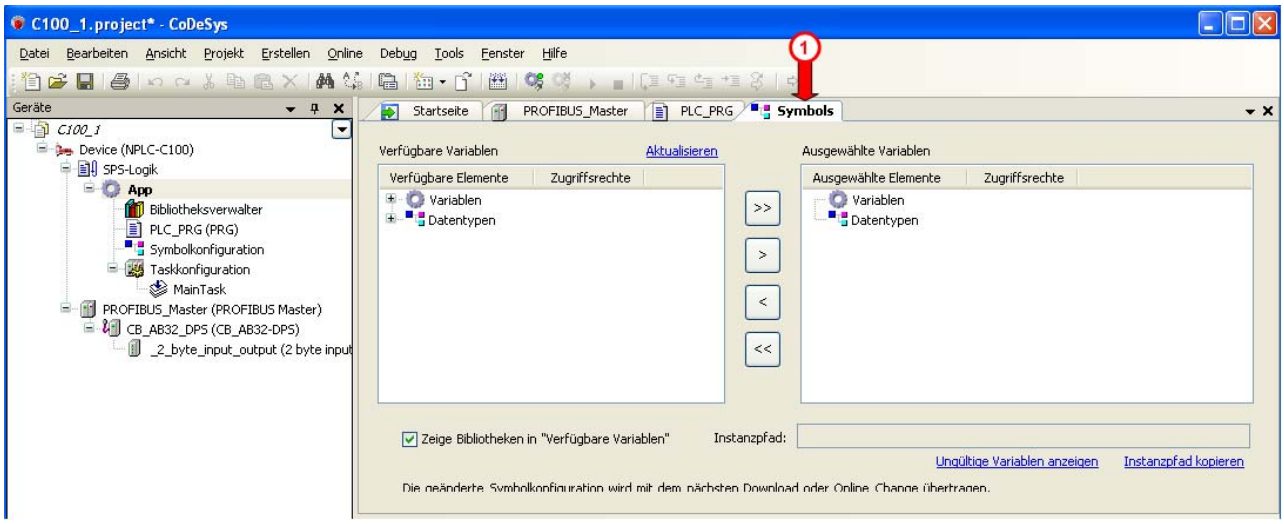



*Abbildung 31: Editor Symbols*

3. Verfügbare Variablen aktualisieren.

➢ Klicken Sie bei **Verfügbare Variablen** auf **Aktualisieren**.

➢ Die verfügbaren Variablen werden angezeigt.


© Hilscher, 2009 - 2014

4. Verfügbare Variablen anzeigen.

   Die verfügbaren Variablen werden in der folgenden Abbildung im Fensterbereich Ⓑ angezeigt.

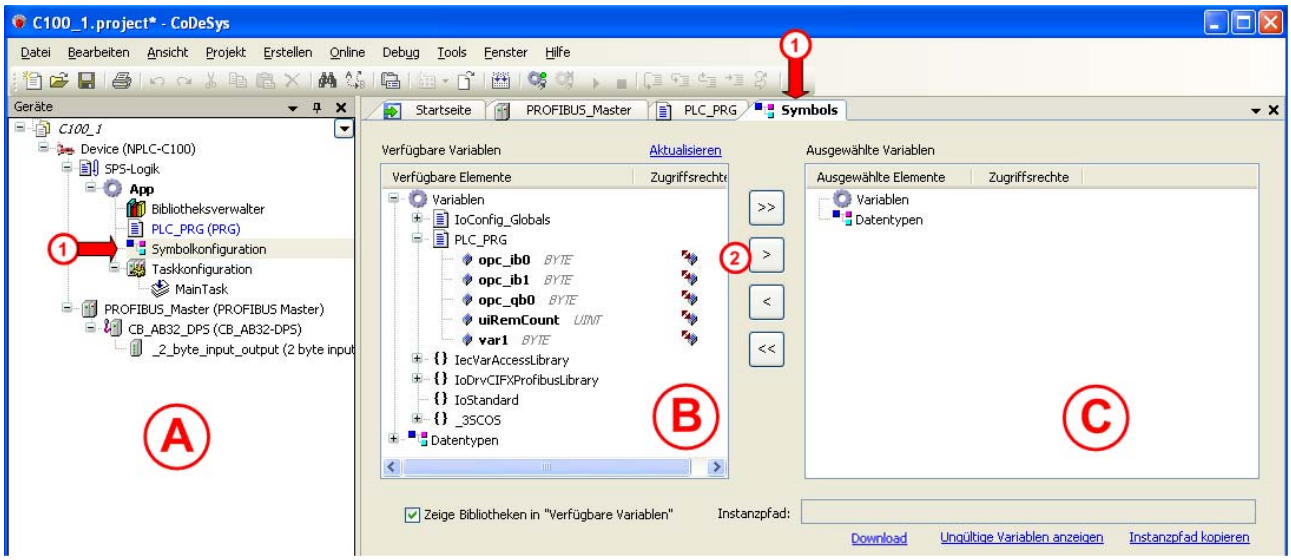



*Abbildung 32: Editor Symbols (2)*

5. Symbole zu den ausgewählten Variablen hinzufügen.

- Markieren Sie im Baum unter **Verfügbare Variablen** Ⓑ die Variable opc_ib0.
- Fügen Sie diese Variable mit einem Klick auf „>“ ② zu den **Ausgewählten Variablen** hinzu.
- Markieren Sie im Baum unter **Verfügbare Variablen** Ⓑ die Variable opc_ib1.
- Fügen Sie diese Variable mit einem Klick auf „>“ ② zu den **Ausgewählten Variablen** hinzu.
- Markieren Sie im Baum unter **Verfügbare Variablen** Ⓑ die Variable opc_qb0.
- Fügen Sie diese Variable mit einem Klick auf „>“ ② zu den **Ausgewählten Variablen** hinzu.


© Hilscher, 2009 - 2014

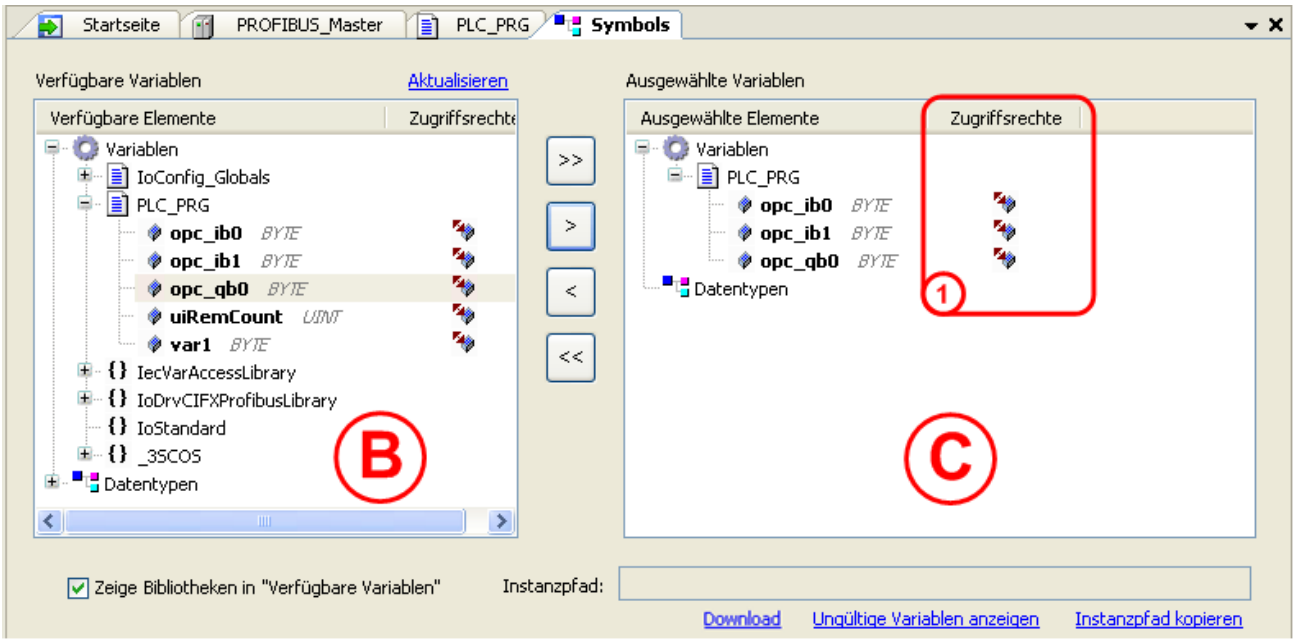



*Abbildung 33: Editor Symbols Zugriffrechte*

Durch Anklicken der Symbole in der Spalte **Zugriffsrechte** im Fensterbereich Ⓒ lassen sich die Lese-/ Schreib-Richtung festlegen.


© Hilscher, 2009 - 2014

### 4.6.2 Projekt übersetzen

Vor dem Laden der Programmierung in die netPLC-Komponente muss das Programm übersetzt werden.

1. Projekt übersetzen.

➢ Wählen Sie **Erstellen > App > Übersetzen F11,** um das Projekt zu übersetzen.

Das Projekt wird übersetzt.

2. Meldungsfenster überprüfen.

➢ Prüfen Sie, ob im Meldungsfenster die Meldung **Übersetzen abgeschlossen – 0 Fehler, 0 Warnungen** erscheint:

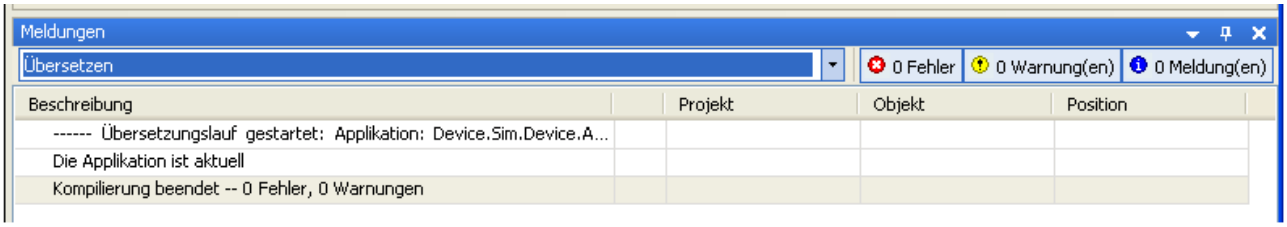



*Abbildung 34: Meldungen nach Übersetzung*

Das Projekt wurde übersetzt.

3. Projekt speichern (zwischenspeichern).

➢ Wählen Sie **Datei > Speichern**, um das Projekt zu speichern.

Das Projekt wird gespeichert.


© Hilscher, 2009 - 2014

# 4.7 Verbindung zur Steuerung herstellen

## 4.7.1 Gateway festlegen

1. Das Fenster Kommunikationseinstellungen öffnen.

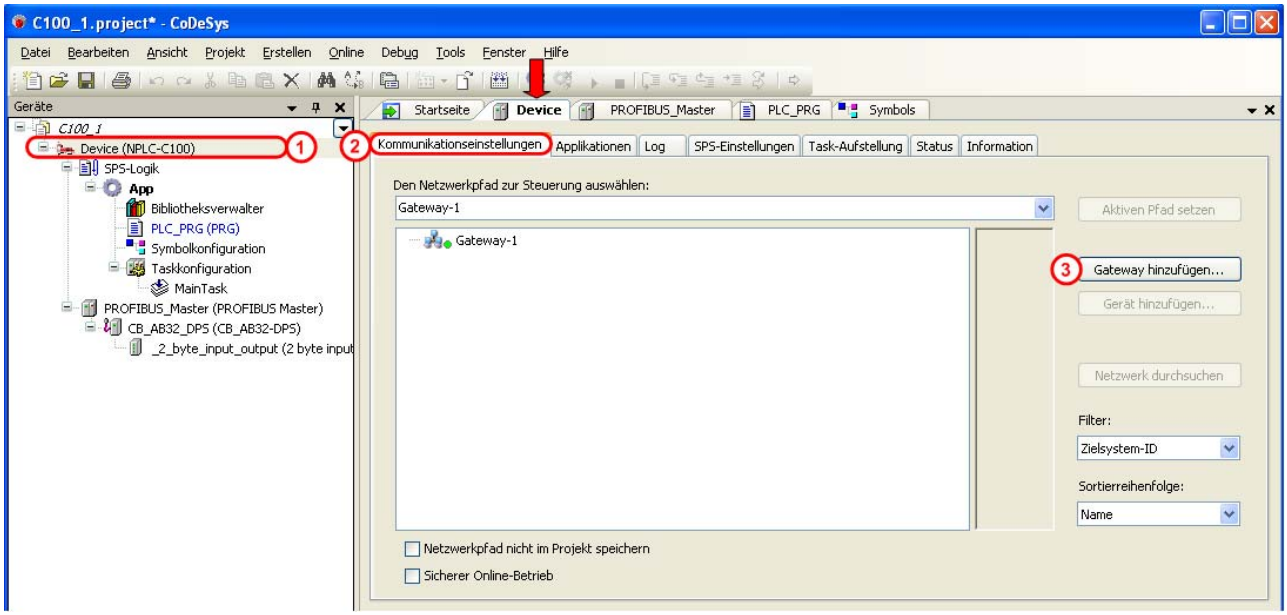



*Abbildung 35: Kommunikationseinstellungen*

➢ Öffnen Sie mit einem Doppelklick auf **Device (NPLC-C100)** ① das Fenster **Device** mit dem Tab **Kommunikationseinstellungen** ②.

➢ Das Fenster **Device** mit dem Tab **Kommunikationseinstellungen** erscheint. (Obige Abbildung)

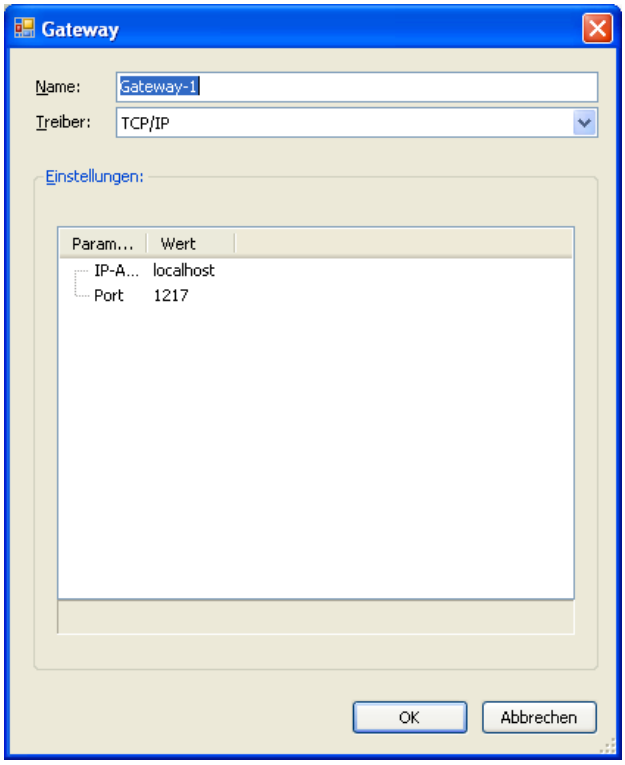
Wenn noch kein Gateway vorhanden ist, dann führen Sie Schritt 2 und 3 aus. Wenn schon ein Gateway vorhanden ist, dann überspringen Sie Schritt 2 und 3.

2. Gateway hinzufügen.

➢ Klicken Sie **Gateway hinzufügen** ③.


© Hilscher, 2009 - 2014

Das Fenster **Gateway** erscheint.




3. Gateway übernehmen.

➢ Klicken Sie auf **OK**, um die Einstellung **Gateway-1** zu übernehmen.

Das Fenster Gateway schließt sich.

## 4.7.2 Netzwerk durchsuchen und Aktiven Pfad setzen

**Hinweis:** Stellen Sie sicher, dass der netPLC CoDeSys Server läuft, wenn mit der Slot-SPS NPLC-C100 über den parallelen PC Systembus (PCI-Bus) kommuniziert wird.

**Hinweis:** Wenn Sie eine Verbindung über USB zur netPLC-Komponente verwenden wollen, dann prüfen Sie, ob das USB Kabel verbunden ist.

Wenn Sie eine Verbindung über Ethernet zur netPLC-Komponente herstellen möchten (z. B. bei netHMI PLC-Geräten), stellen Sie sicher, dass eine Ethernet-Verbindung zu dem Gerät besteht.

1. Netzwerk durchsuchen.

➢ Klicken Sie auf die Schaltfläche **Netzwerk durchsuchen**, um nach SPS Geräten zu suchen.

Geräte werden über die USB-Verbindung, über den netPLC CoDeSys Server sowie über Ethernet gesucht.


© Hilscher, 2009 - 2014

- Gefundene Geräte (auch netPLC artfremde Geräte) werden unterhalb von **Gateway-1** angezeigt.

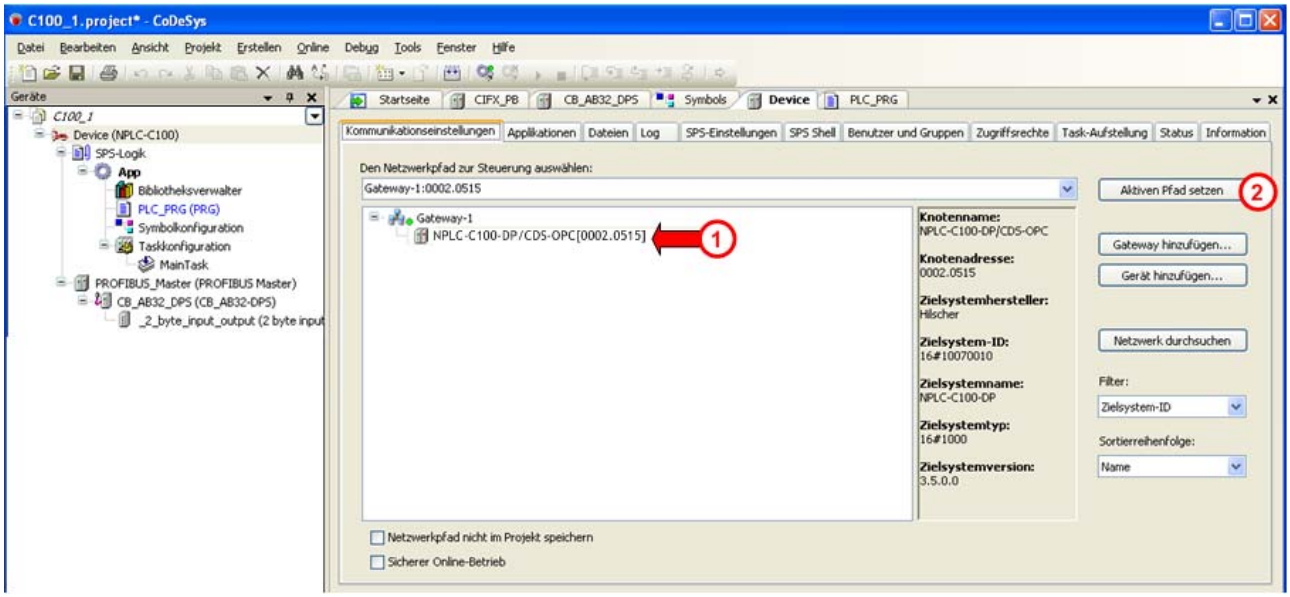



*Abbildung 36: Aktiven Pfad setzen*

2. Aktiven Pfad setzen.

- Klicken Sie unterhalb von **Gateway-1** auf den **NPLC-Eintrag** ①, um das Gerät zu markieren.
- Klicken Sie auf die Schaltfläche **Aktiven Pfad setzen** ②, um eine Verbindung zum Gerät zu aktiveren.

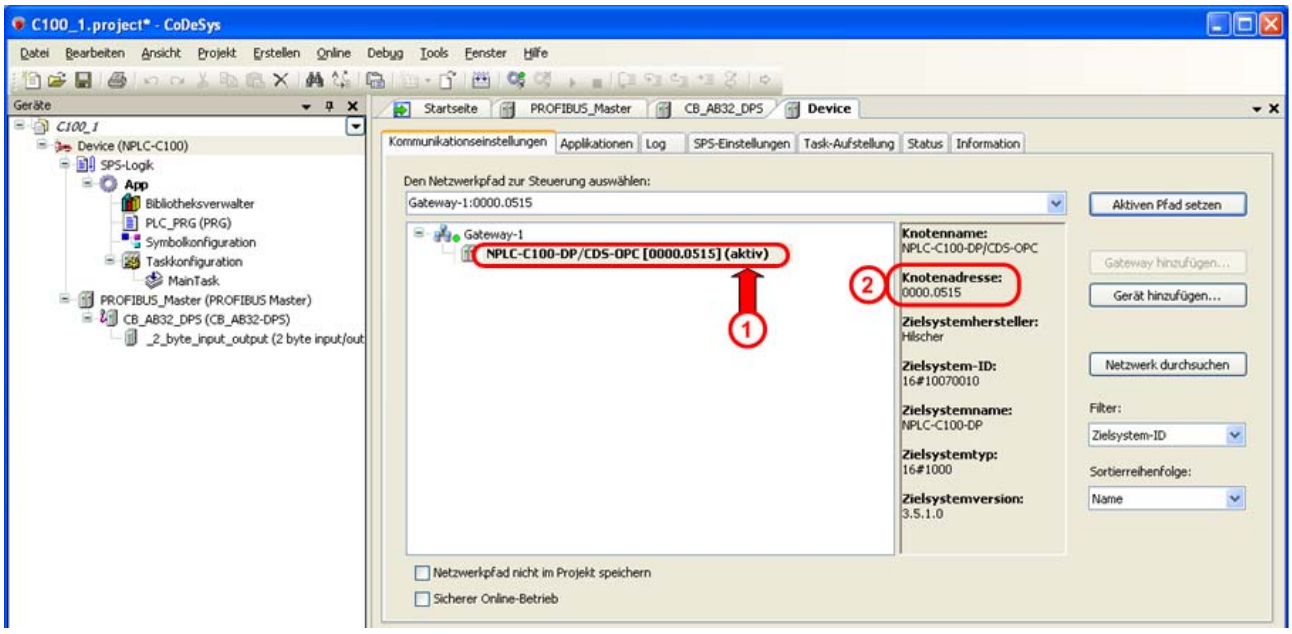



*Abbildung 37: Pfad auf aktiv gesetzt*

- Neben dem Gerät **NPLC-C100** erscheint die Angabe **aktiv** ①.

  **Hinweis:** Die angezeigte **Knotenadresse** ② benötigen Sie, wenn Sie den OPC-Server konfigurieren.

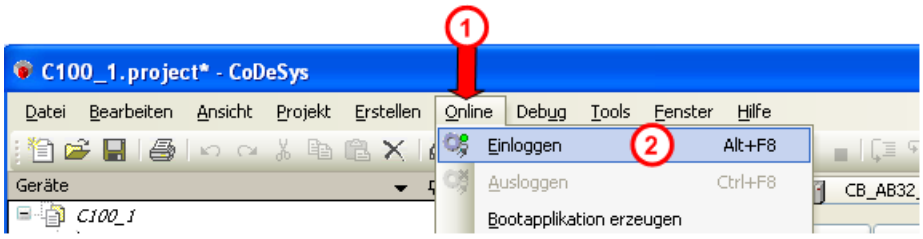

© Hilscher, 2009 - 2014

# 4.8 Steuerungsprogramm und Buskonfiguration übertragen

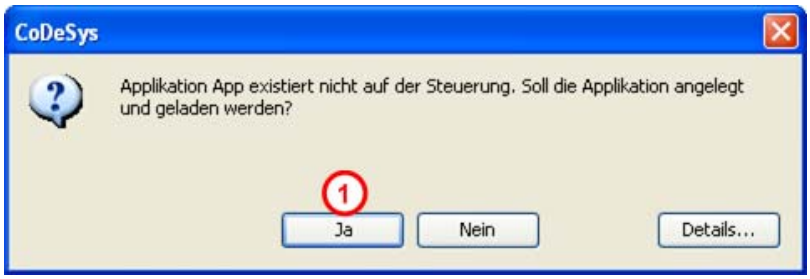
## 4.8.1 Einloggen und übertragen

1. Auf der Steuerung einloggen.

➢ Wählen Sie **Online > Einloggen**.




Die Software überprüft, ob das Projekt auf der Zielhardware identisch zu dem aktuell geöffneten Projekt auf dem PC ist. Befindet sich auf der Zielhardware noch kein Programm, dann erfolgt die Meldung „Applikation App existiert nicht auf der Steuerung. Soll die Applikation geladen werden?“. Sind die Projekte unterschiedlich, so erfolgt die Meldung „Warnung: Eine unbekannte Version von Applikation ‚App’ läuft gerade auf der Steuerung. Möchten Sie dennoch den aktuellen Code laden und die bestehende Applikation ersetzen?“.




**Wichtig:** Ein Laden des SPS-Programms nur über das Einloggen speichert das SPS-Programm nicht remanent in der netPLC-Komponente. Sollte die Baugruppe spannungslos gemacht werden, ist das zunächst in den flüchtigen Speicher übertragene SPS-Programm und die Buskonfiguration gelöscht. Es kann daher passieren, dass ein vorher in den remanenten FLASH-Speicher der Zielhardware geladenes altes Programm wieder gestartet wird.

Damit das übertragene SPS-Programm und Buskonfiguration spannungsausfallsicher in der netPLC-Komponente gespeichert wird, ist es zwingend notwendig, dass ein Bootprojekt erzeugt wird. Dies wird in Abschnitt *Bootprojekt erzeugen* ab Seite 53 beschrieben.

2. Ladevorgang starten.

➢ Beantworten Sie die Frage „Applikation App existiert nicht auf der Steuerung. Soll die Applikation geladen werden?“ mit **Ja** (1).

Die Applikation wird auf die Steuerung geladen.

Beim Start des Ladens geht die SPS in den STOP-Modus.


© Hilscher, 2009 - 2014

- Die Fußzeile zeigt den Fortschritt des Downloadvorganges an: **Sending download info: Laden …** und die Angabe der Größe des bereits geladenen Programms sowie die Gesamtgröße.

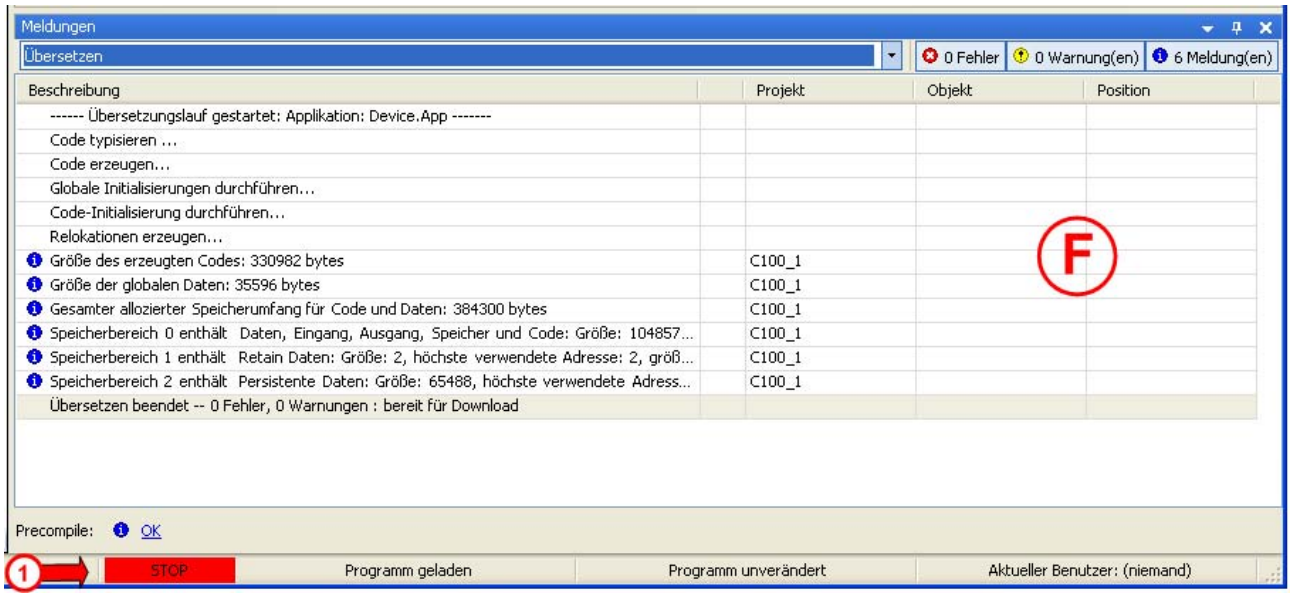



*Abbildung 38: Meldungen zur Programmübersetzung*

- Nach dem Ladevorgang zeigt die Statuszeile den Zustand des geladenen Programms und der Steuerung an.

**Wichtig:** Die Steuerung bleibt im Zustand Stop. Dieses ist in der Statuszeile bei ① zu sehen. Dieses wird ebenfalls im Gerätebaum in der Zeile **App** signalisiert.

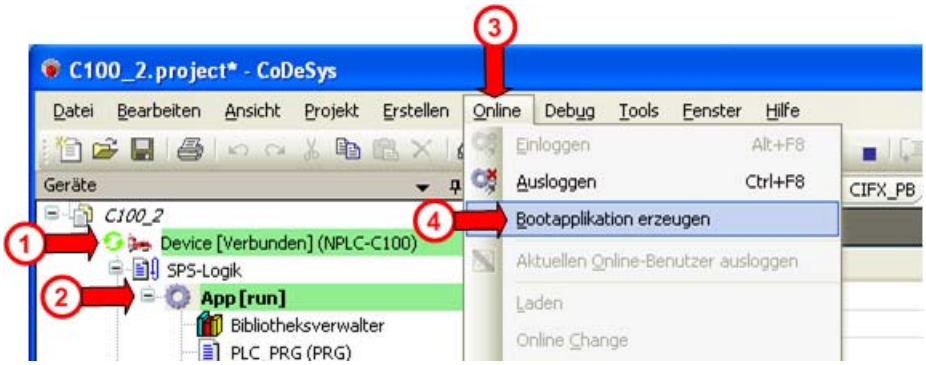

© Hilscher, 2009 - 2014

## 4.8.2 Bootprojekt erzeugen

**Wichtig:** Damit das SPS-Programm und die Buskonfiguration spannungsausfallsicher in der netPLC-Komponente gespeichert wird, ist es zwingend notwendig, dass ein Bootprojekt erzeugt wird.

**Hinweis:** Sie können nur ein Bootprojekt auf der Baugruppe erzeugen, wenn Sie mit der netPLC-Komponente online verbunden (Eingeloggt) sind. Andernfalls werden Sie gefragt, wo Sie das Bootprojekt im PC System speichern möchten.

Bootprojekt erzeugen.




➢ Vergewissern Sie sich, dass das Programmiersystem eingeloggt ist. Dieses ist an den grün hinterlegten Zeilen ① und ② zu erkennen.

Es ist dabei unerheblich, ob die Applikation (das SPS-Programm) in **[Run]** oder **[Stop]** steht.

➢ Wählen Sie **Online** ③ **> Bootapplikation erzeugen** ④.

↬ Das Bootprojekt wird erzeugt und in die Steuerung geladen.

## 4.8.3 Bootprojekt entfernen

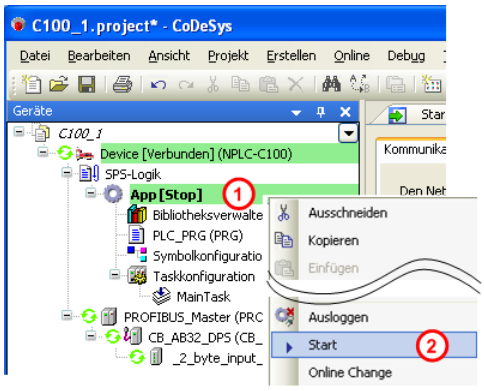
Bootprojekt entfernen.

➢ Falls ein vorhandenes Bootprojekt entfernt werden soll, wählen Sie **Online > Reset Ursprung**.

↬ Die netPLC-Komponente wird zurückgesetzt und das vorhandene Bootprojekt gelöscht.


© Hilscher, 2009 - 2014

### 4.8.4 Steuerungsprogramm starten

➢ 


Wählen Sie mit der rechten Maustaste das Kontextmenü des Gerätebaums der Zeile **App** (1), aus. Wählen Sie im Kontextmenü den Eintrag **Start** (2) aus, um das SPS-Programm zu Starten.

➢ **Wichtig:** Prüfen Sie, ob ein eventuell vorhandener Betriebswahlschalter der netPLC-Komponente in der Stellung RUN steht. Wenn der Schalter in der Stellung STOP steht, dann schalten Sie ihn um auf RUN, ansonsten geht die SPS nicht in den Zustand [Run].

➢ Die Steuerung geht in den Zustand RUN.
Dieses wird Ihnen in der Statuszeile des Programms durch LÄUFT (4) angezeigt.

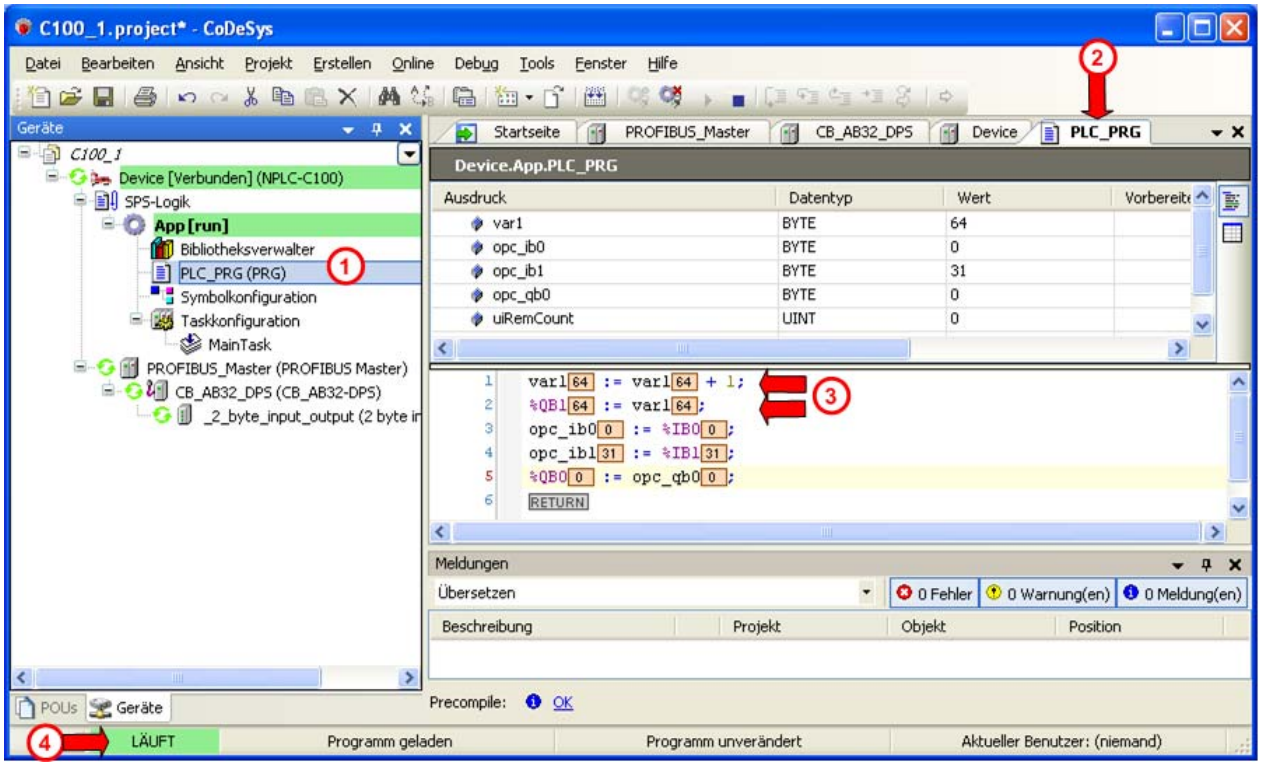



*Abbildung 39: Gestartetes SPS-Programm*

➢ Machen Sie mit der Maus einen Doppelklick auf PLC_PRG (1), oder wählen Sie den Reiter **PLC_PRG** (2).

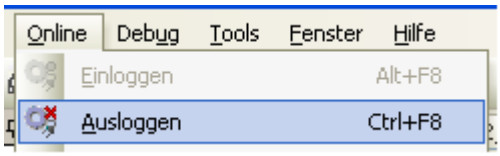
➢ In den mit (3) markierten Zeilen können Sie das Hochzählen der Variablen beobachten. Das SPS-Programm erhöht mit jedem SPS Zyklus das Ausgangsbyte %QB1 und 1. Prüfen Sie, dass am PROFIBUS-DP Slave sich dessen Ausgangsdaten ändern. Verändern Sie am PROFIBUS DP Slave dessen Eingangsdaten und prüfen Sie, ob diese in der Steuerung ‚ankommen'.


© Hilscher, 2009 - 2014

# 4.9 Verbindung zur Steuerung aufheben

Ausloggen

➢ Wählen Sie
**Online > Ausloggen Ctrl+F8**,
um sich auszuloggen.




➢ Die Verbindung zur Steuerung wird aufgehoben.


© Hilscher, 2009 - 2014

# 5 Visualisierung

## 5.1 Visualisierung HMI-Steuerung netHMI PLC

Die Visualisierung der Steuerungs-Daten erfolgt bei den netHMI PLC-Geräten (HMI-Steuerungen) **NHMI J500Q5-DP-DP/DPM/CDS/QV** und **NHMI J500Q5-RE-DP/PNM/CDS/QV** mit der QViS Visualisierungssoftware von Hilscher direkt auf den Geräten selbst. Detaillierte Informationen hierzu finden Sie im Benutzerhandbuch *netHMI und QViS*, DOC120703UMxxDE, das Sie auf der netHMI PLC Produkt-DVD im Verzeichnis `Documentation\deutsch\1.Software` finden.

## 5.2 Visualisierung Slot-SPS NPLC-C100 und Modul-SPS NPLC-M100

Daten aus diesen Steuerungen können über den OPC-Server von einer Visualisierung mit OPC Schnittstelle (im folgenden OPC-Klient genannt) gelesen bzw. geschrieben werden.

Dazu ist eine Konfiguration notwendig. Die folgenden Abschnitte beschreiben die Konfigurationsschritte, um eine Visualisierung über OPC mit der Steuerung **NPLC-C100** bzw. **NPLC-M100** zu verbinden.

**Voraussetzungen**

Betriebsbereite Steuerung, bestehend aus:

- Installierte Hardware
- Installierter Gerätetreiber: cifX Device Driver
- Betriebsbereiter Server: netPLC CoDeSys Server
- Die Steuerung wurde bereits mit CoDeSys programmiert
- Betriebsbereite(s) Feldbus Slave Gerät(e), die über ein Feldbus-Kabel mit der Steuerung verbunden sind

Installierter CoDeSys OPC Server

- Installierter 3S OPC Server

Der netPLC CoDeSys Server muss auf dem PC des laufenden OPC-Servers gestartet sein und Zugriff auf die Slot-SPS(en) haben.


© Hilscher, 2009 - 2014

### 5.2.1 OPC Konfigurator

1. OPC Konfigurator starten.

➢ Starten Sie den OPC Konfigurator **Start > Alle Programme > 3S CoDeSys > CoDeSys OPC Server 3 > OPC Configurator**.

↬ Die Oberfläche des OPC Konfigurators erscheint.

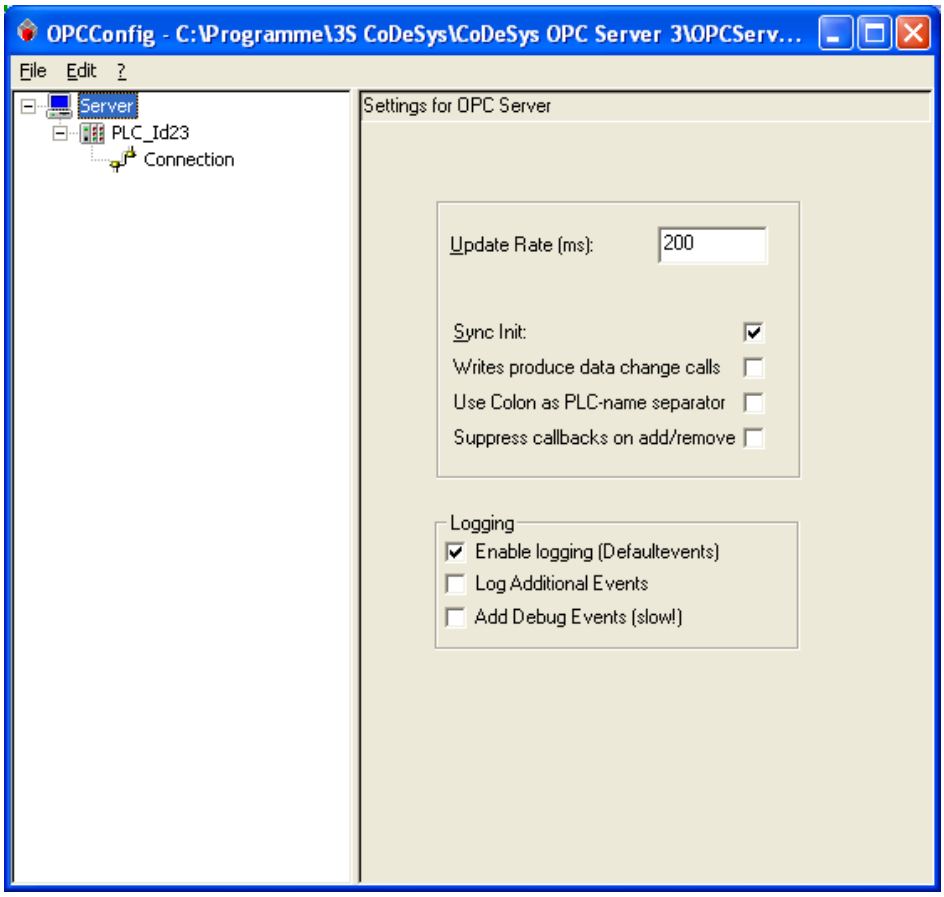



*Abbildung 40: OPC-Server Konfigurationseinstieg*

➢ Löschen Sie den Eintrag PLC_Id23 über das Kontextmenü.

2. PLC einfügen.

➢ Wählen Sie **Edit > Append PLC** oder die Tastenkombination Strg+A.

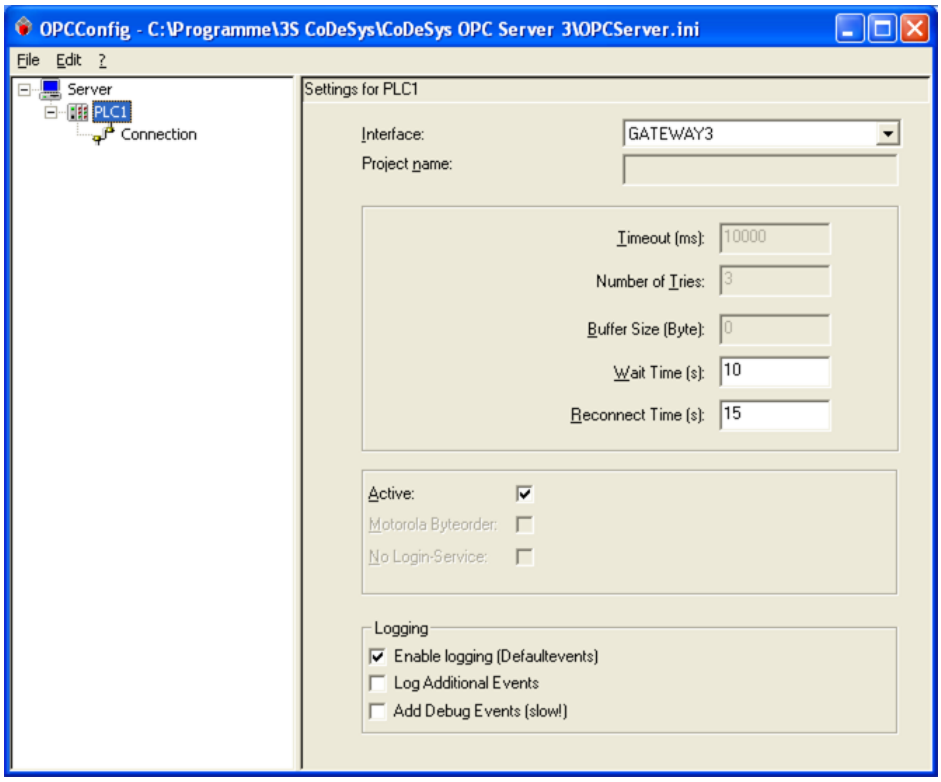
↬ Das Fenster **Settings for PLC** erscheint.

↬ Der Defaultname lautet **PLC1**.


© Hilscher, 2009 - 2014

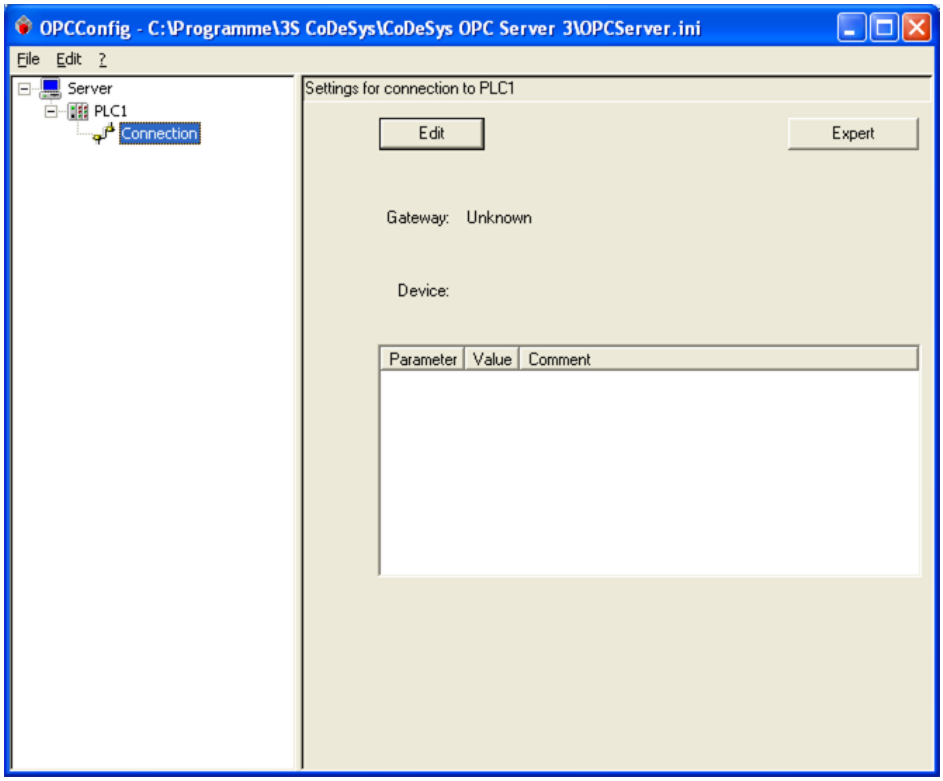
3. Das Interface auf GATEWAY3 einstellen.

➢ Stellen Sie das **Interface** auf **GATEWAY3**.




4. **Connection** bearbeiten.

➢ Wählen Sie in der Baumdarstellung den Eintrag **Connection**.





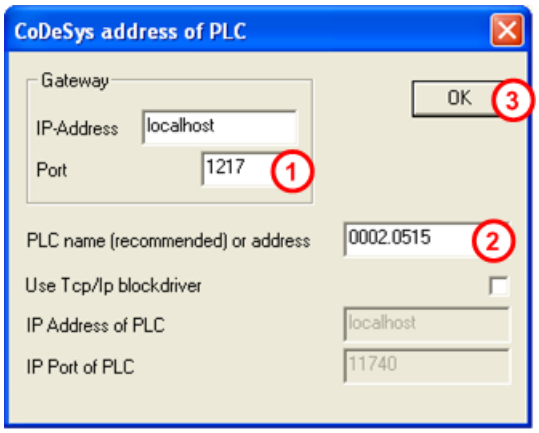
© Hilscher, 2009 - 2014

5. Einstellungen bearbeiten.

➢ Wählen Sie **Edit**.

↬ Das Fenster zur Adress-Eingabe erscheint.

6. CoDeSys PLC Adresse eingeben.




➢ Verwenden Sie die **Port**nummer (1) 1217.

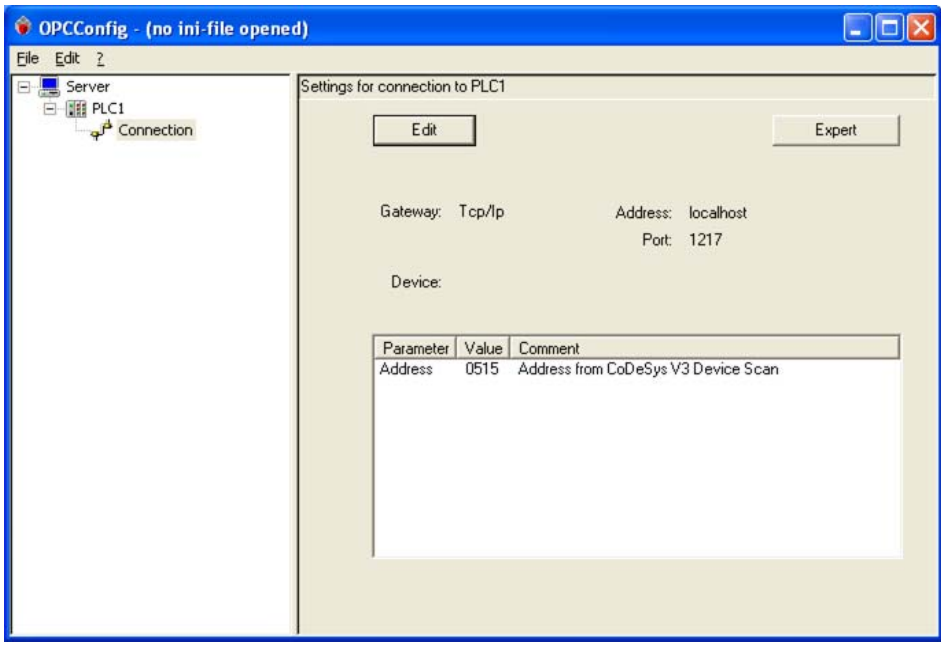
➢ Geben Sie bei **PLC name or address** (2) die Adresse ein, über die das 3S Gateway auf die Slot-SPS NPLC-C100 zugreifen kann, z. B. 0002.0515.

**Wichtig:** Verwenden Sie die Knotenadresse aus der CoDeSys Programmiersoftware, die angezeigt wird, wenn der Aktive Pfad gesetzt wird. Siehe (2) in der *Abbildung 37: Pfad auf aktiv gesetzt* Seite 50.

7. Schließen Sie das Fenster.

➢ Klicken Sie auf **OK** (3), um das Fenster zu schließen.

↬ Die Einstellungen werden zwischengespeichert.



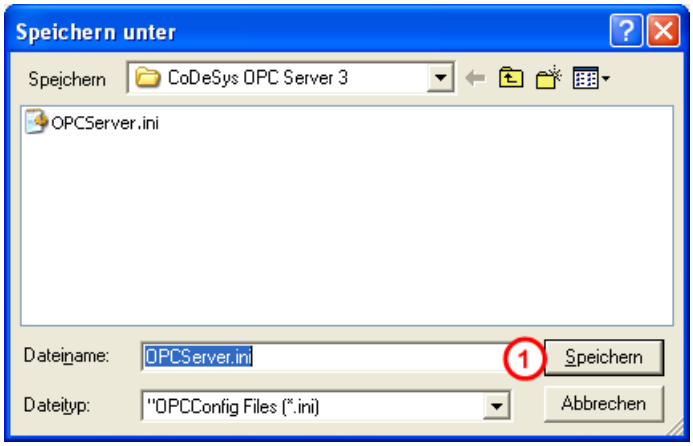


© Hilscher, 2009 - 2014

8. Konfiguration speichern.

➢ Wählen Sie **File > Save As**.

➢ Geben Sie bei Dateiname den Namen **OPCServer.ini** ein und überschreiben die vorhandene Datei.




➢ Klicken Sie auf **Speichern** (1).

➢ Die OPC Konfigurationsdatei wird gespeichert.

9. OPC Konfigurator beenden.

➢ Wählen Sie **File > Exit**, um den OPC Konfigurator zu beenden.


© Hilscher, 2009 - 2014

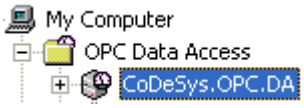
## 5.2.2 OPC-Klient

Der Zugriff von einem OPC-Klient kann hier nur allgemein wiedergegeben werden, da dies von der verwendeten Visualisierungssoftware abhängt.

1. Starten Sie OPC-Klient Software.

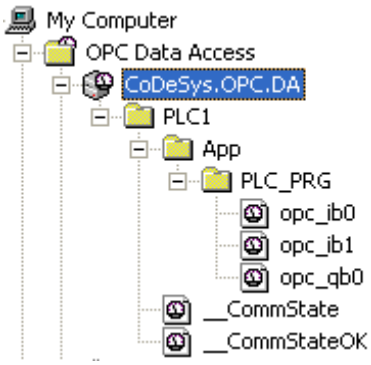
2. Stellen Sie eine Verbindung zum OPC Server her.

↬ Der CoDeSys OPC Server startet automatisch als Hintergrundprozess auf dem Ziel-PC, in dem die Slot-SPS NPLC-C100 installiert ist.




3. Die Variablen werden eingelesen und können angesehen werden.

↬




4. Fügen Sie beim OPC-Klient eine Gruppe hinzu.

5. Fügen Sie in diese Gruppe die Items opc_ib0, opc_ib1 und opc_qb0 hinzu.

6. Werte werden nun von der Steuerung gelesen.

7. Werte können an die Steuerung geschrieben werden: opc_qb0 kann geschrieben werden.

↬

| Name | Value | Quality |
|---|---|---|
| PLC1.App.PLC_PRG.opc_ib0 | 0 (VT_UI1) | Good - Non-spe... |
| PLC1.App.PLC_PRG.opc_ib1 | 255 (VT_UI1) | Good - Non-spe... |
| PLC1.App.PLC_PRG.opc_qb0 | 13 (VT_UI1) | Good - Non-spe... |


© Hilscher, 2009 - 2014

# 6 Host-E/A-Schnittstelle (Slot-SPS NPLC-C100)

Bei der Slot-SPS NPLC-C100 kann das Anwendungsprogramm am Host auf Eingangs- und Ausgangsdaten der SPS direkt zugreifen. Im CoDeSys-Projekt muss dazu ein **Host I/O Interface** als ‚Gerät' in das Projekt eingefügt werden. Die Vorgehensweise dazu ist in Abschnitt *Host E/As bei NPLC-C100 Slot-SPS* auf Seite 19 beschrieben. Das Anwendungsprogramm nutzt die Funktionen des cifX Device Driver, um Eingangsdaten zu lesen und Ausgangsdaten zu schreiben.

Die folgende Abbildung zeigt den prinzipiellen Aufbau.

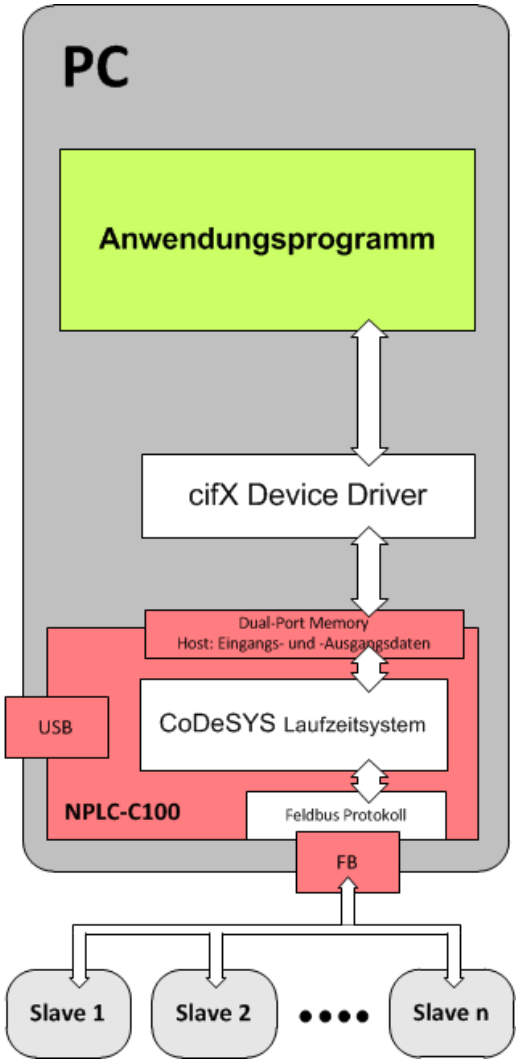



*Abbildung 41: Host-E/A-Schnittstelle*

Wenn das **Host I/O Interface** als erstes ‚Gerät' in das SPS-Projekt eingefügt wird und danach die Slave-Geräte, dann belegt das **Host I/O Interface** im SPS-Speicher die Adressen %IB0 … %IB511 und %QB0 … %QB511.

Wenn das Host I/O Interface nach den Slave-Geräten in das SPS-Projekt eingefügt wird, dann belegt das **Host I/O Interface** im SPS-Speicher die nächsten freien Adressen. Belegten die Slave-Geräte beispielsweise bereits die Adressen %IB0 … %IB15 und %QB0 … %QB15, dann belegt das Host I/O Interface die Adressen %IB16 … %IB527 und %QB16 … %QB527.

Im Speicher zum Host liegen die Ein- und Ausgangsdaten (für den cifX Device Driver) immer auf den Offsets 0 bis 511.


© Hilscher, 2009 - 2014

# 7 Funktionen

## 7.1 Funktionsbibliothek erweitern

Bevor Sie die in diesem Abschnitt dokumentierten Funktionen verwenden können, müssen Sie diese erst in die Funktionsbibliothek von CoDeSys einfügen.

Hierzu gehen Sie wie folgt vor:

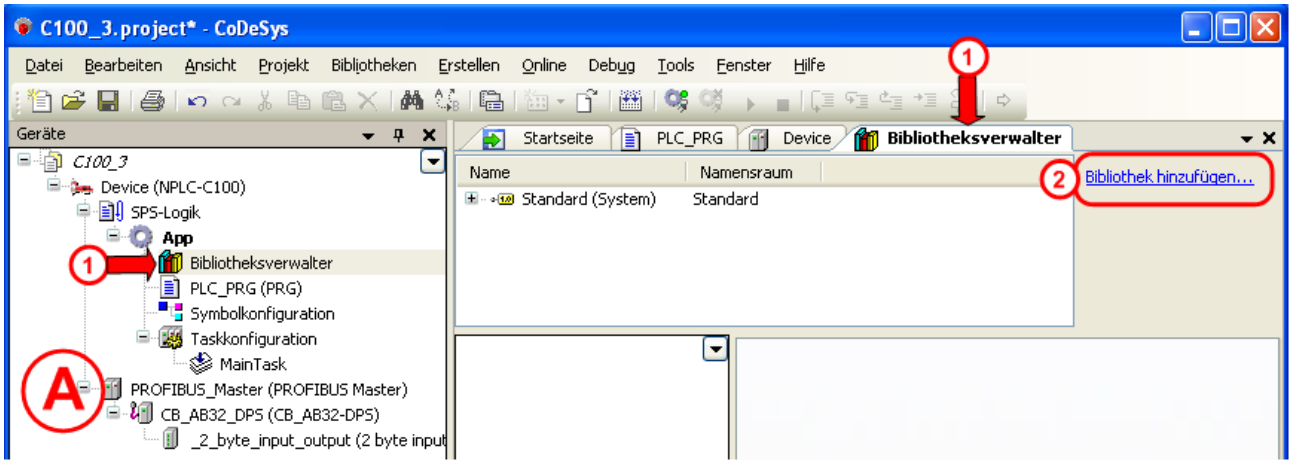
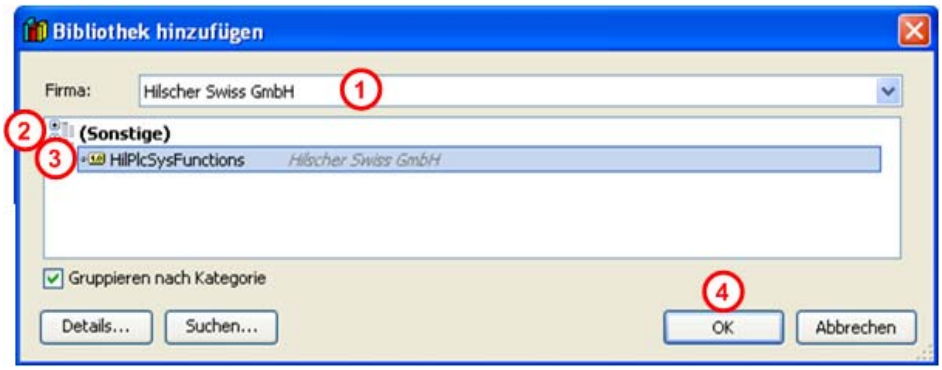



*Abbildung 42: Bibliothek hinzufügen*

1. Machen Sie im Gerätebaum (A) einen Doppelklicken auf den Eintrag (1) **Bibliotheksverwalter**.

➪ Es öffnet sich folgendes Popup:




2. Wählen Sie unter **Firma** (1) > **Hilscher Swiss GmbH** aus.
- ➢ Machen Sie einen Doppelklick vor den Eintrag **(Sonstige)** (2).
- ➢ Klicken Sie zum Markieren auf die Zeile **HilPlcSysFunktions** (3).
- ➢ Klicken Sie **OK** (4).


© Hilscher, 2009 - 2014

➢ Die Bibliotheks-Verwaltungsübersicht hat sich wie folgt verändert:

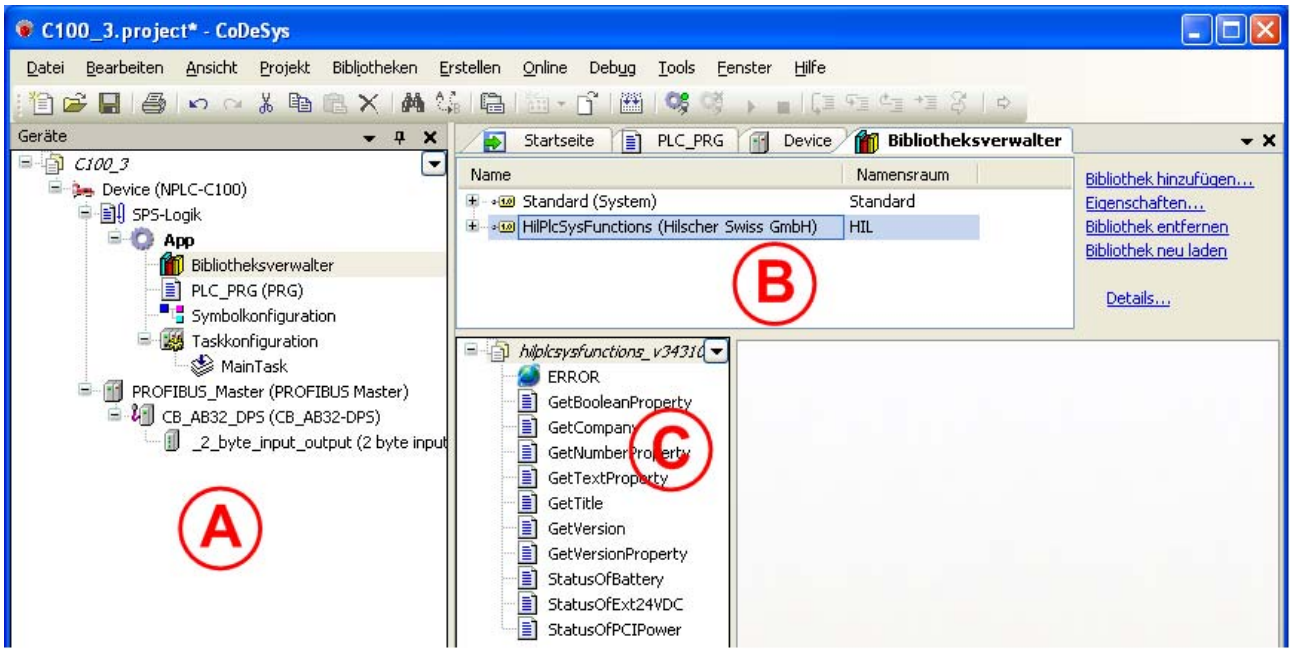



*Abbildung 43: Bibliothek hinzufügen*

Im Fensterbereich Ⓑ sehen Sie die integrierten Bibliotheken und im Fensterbereich Ⓒ die Funktionen der im Fensterbereich Ⓑ ausgewählten Bibliothek.

## 7.2 Funktionen – HilPlcSysFunctions

Die Funktionsbibliothek HilPlcSysFunctions bietet drei Funktionen für die netPLC-Komponenten:

| Systemfunktionsname | Funktion |
|---|---|
| StatusOfBattery | Prüft den Status der Batterie und misst die Batteriespannung. |
| StatusOfExt24V | Prüft, ob die netPLC Komponente über die externe Spannungsversorgung mit Spannung versorgt wird. |
| StatusOfPCIPower | Prüft, ob die netPLC Komponente über die PCI Spannungsversorgung mit Spannung versorgt wird. |

*Tabelle 8: Systemfunktionen – CmpSysFunctions_netPLC*

Zum Einfügen dieser Funktionen in das SPS-Progamm kann die Eingabehilfe verwendet werden.

Zum Öffnen der Eingabehilfe …

- ➢ Drücken Sie F2 (oder)
- ➢ Wählen Sie das Menü **Bearbeiten > Eingabehilfe** (oder)
- ➢ Wählen Sie im Editor aus dem Kontextmenü den Eintrag **Eingabehilfe**.


© Hilscher, 2009 - 2014

Es öffnet sich das Popup der Eingabehilfe.

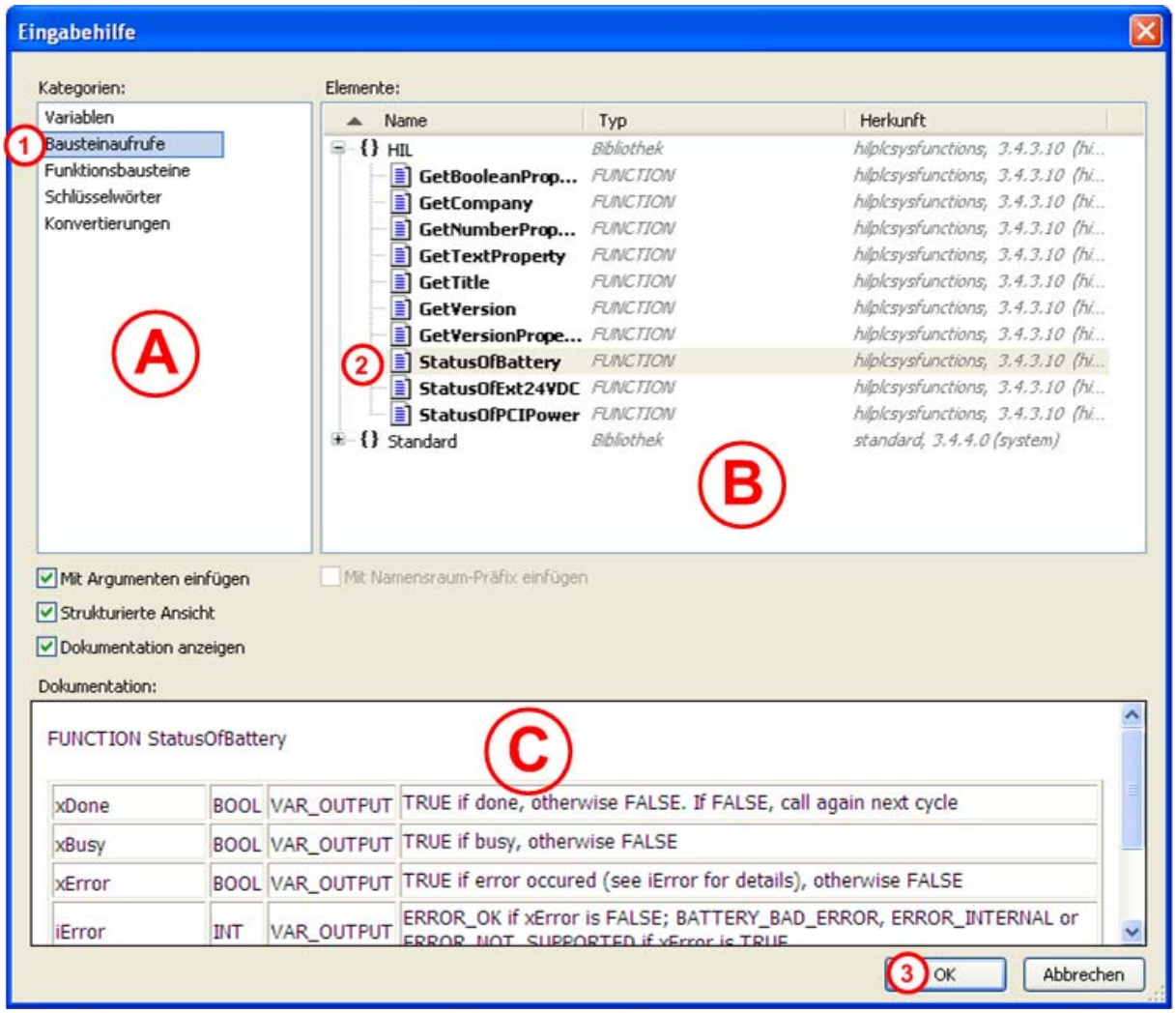



*Abbildung 44: Eingabehilfe Bausteinaufrufe*

- Wählen Sie dann unter **Kategorien** Ⓐ den Eintrag **Bausteinaufrufe** ①.
- Wählen Sie unter **Elemente** Ⓑ z.B. den Eintrag **StatusOfBattery**.
- Im Fensterbereich Ⓒ wird die Dokumentation zur Systemfunktion angezeigt.
- Klicken Sie auf **OK** ③, um die Funktion in den PSP-Code zu übernehmen.


© Hilscher, 2009 - 2014

## 7.2.1 Batterie Status – StatusOfBattery

*Abbildung 45: Systemfunktion StatusOfBattery*

| **Funktion** | **StatusOfBattery** | | | |
|---|---|---|---|---|
| Beschreibung | Prüft den Status der Batterie und misst die Batteriespannung.<br>Achtung: Bei der Prüfung des Batteriestatus wird die Batterie belastet, d. h. es findet eine geringe Entladung statt. Diese Funktion deshalb nicht zyklisch aufrufen.<br>Der dabei angesprochene A/D-Wandler benötigt einige Zeit für die Messung. Daher ist für die Funktion eine State Machine erforderlich. Siehe Beispiel im Abschnitt *Beispiel SPS-Programm zur Batterie-Überwachung* (NPLC-C100 auf Seite 70. | | | |
| **Name** | **Datentyp** | **Ein-/Ausgang** | **Initialwert** | **Kommentar** |
| xDone | BOOL | Ausgang | FALSE | TRUE: Systemfunktion fertig<br>FALSE: Diese Systemfunktion im nächsten SPS Zyklus wieder aufrufen |
| xBusy | BOOL | Ausgang | FALSE | TRUE: Systemfunktion in Bearbeitung<br>FALSE: Systemfunktion nicht in Bearbeitung |
| xError | BOOL | Ausgang | FALSE | TRUE: Fehler aufgetreten. Die Fehlernummer wird in iError ausgegeben.<br>FALSE: Kein Fehler aufgetreten |
| iError | INT | Ausgang | 0 | Wenn xError den Wert FALSE hat, dann<br>NO_ERROR: Kein Fehler (0)<br>Wenn xError den Wert TRUE hat, dann<br>BATTERY_BAD_ERROR: (1001)<br>INTERNAL_ERROR: (1000)<br>NOT_SUPPORTED: (1010) (Messung wird von der Hardware nicht unterstützt.) |
| xStatusOfBattery | BOOL | Ausgang | FALSE | TRUE: Batteriestatus “Gut”<br>FALSE: Batteriestatus “Schlecht”: (Batteriespannung < 2,75V) → Batterie wechseln |
| BatteryVoltage | REAL | Ausgang | 0 | Gemessener Spannungswert der Batterie in V.<br>Ein Spannungswert über 2,75 V bedeutet, dass die Batteriespannung ausreichend ist. |

*Tabelle 9: Systemfunktion StatusOfBattery*


© Hilscher, 2009 - 2014

#### 7.2.1.1 Verhalten der Funktion in netPLC-Komponenten, die keine Hardware-Batterie-Spannungsüberwachung haben.

In diesen Komponenten führt die Funktion zu keinem fatalen Fehler.

Die Funktion kommt sofort mit folgenden Werten zurück:

| Name | Datentyp | Ein-/Ausgang | Initialwert | Kommentar |
|---|---|---|---|---|
| xDone | BOOL | Ausgang | FALSE | TRUE: Systemfunktion fertig |
| xBusy | BOOL | Ausgang | FALSE | FALSE: Systemfunktion nicht in Bearbeitung |
| xError | BOOL | Ausgang | FALSE | TRUE: Fehler aufgetreten. Die Fehlernummer wird in iError ausgegeben. |
| iError | INT | Ausgang | 0 | NOT_SUPPORTED: (1010) (Messung wird von der Hardware nicht unterstützt.) |
| xStatusOfBattery | BOOL | Ausgang | FALSE | TRUE: Batteriestatus “Gut” |
| BatteryVoltage | REAL | Ausgang | 0 | 3 V |

*Tabelle 10: Systemfunktion StatusOfBattery, Hardware ohne AD-Wandler*

## 7.2.2 Status externe Spannungsversorgung – StatusOfExt24VDC

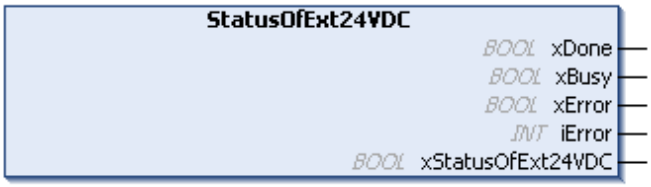



*Abbildung 46: Systemfunktion StatusOfExt24VDC*

| Funktion | StatusOfExt24VDC | | | |
|---|---|---|---|---|
| Beschreibung | Prüft, ob die netPLC Komponente über die externe Spannungsversorgung mit Spannung versorgt wird. | | | |
| **Name** | **Datentyp** | **Ein-/Ausgang** | **Initialwert** | **Kommentar** |
| xDone | BOOL | Ausgang | FALSE | TRUE: Systemfunktion fertig<br>FALSE: Diese Systemfunktion im nächsten SPS Zyklus wieder aufrufen |
| xBusy | BOOL | Ausgang | FALSE | TRUE: Systemfunktion in Bearbeitung<br>FALSE: Systemfunktion nicht in Bearbeitung |
| xError | BOOL | Ausgang | FALSE | TRUE: Fehler aufgetreten. Die Fehlernummer wird in iError ausgegeben.<br>FALSE: Kein Fehler aufgetreten |
| iError | INT | Ausgang | 0 | Wenn xError den Wert FALSE hat, dann<br>NO_ERROR: Kein Fehler (0)<br>Wenn xError den Wert TRUE hat, dann<br>INTERNAL_ERROR: (1000)<br>NOT_SUPPORTED: (1010) |
| xStatusOfExt24VDC | BOOL | Ausgang | FALSE | TRUE: Externe 24 V Spannungsversorgung vorhanden Uext. > 18 V<br>FALSE: Externe 24 V Spannungsversorgung nicht vorhanden Uext. < 18 V |

*Tabelle 11: Systemfunktion StatusOfExt24VDC*


© Hilscher, 2009 - 2014

## 7.2.3 Status Spannungsversorgung über PCI – StatusOfPCIPower

*Abbildung 47: Systemfunktion StatusOfPCIPower*

| **Funktion** | **StatusOfPCIPower** | | | |
|---|---|---|---|---|
| Beschreibung | Prüft, ob die netPLC Komponente über die PCI Spannungsversorgung mit Spannung versorgt wird. | | | |
| **Name** | **Datentyp** | **Ein-/Ausgang** | **Initialwert** | **Kommentar** |
| xDone | BOOL | Ausgang | FALSE | TRUE: Systemfunktion fertig<br>FALSE: Diese Systemfunktion im nächsten SPS Zyklus wieder aufrufen |
| xBusy | BOOL | Ausgang | FALSE | TRUE: Systemfunktion in Bearbeitung<br>FALSE: Systemfunktion nicht in Bearbeitung |
| xError | BOOL | Ausgang | FALSE | TRUE: Fehler aufgetreten. Die Fehlernummer wird in iError ausgegeben.<br>FALSE: Kein Fehler aufgetreten |
| iError | INT | Ausgang | 0 | Wenn xError den Wert FALSE hat, dann<br>NO_ERROR: Kein Fehler (0)<br>Wenn xError den Wert TRUE hat, dann<br>INTERNAL_ERROR: (1000)<br>NOT_SUPPORTED: (1010) |
| xStatusOfPciPower | BOOL | Ausgang | FALSE | TRUE: PCI Spannungsversorgung vorhanden<br>FALSE: PCI Spannungsversorgung nicht vorhanden |

*Tabelle 12: Systemfunktion StatusOfPCIPower*


© Hilscher, 2009 - 2014

## 7.3 Reset-Verhalten (NPLC-C100)

Die Slot-SPS **NPLC-C100** hat keine Reset-Verbindung über den PCI Bus zum PC. D. h. ein Reset des PC beispielsweise über den Reset-Taster wird nicht an die Slot-SPS weitergegeben.

Der PowerOn-Reset wird von beiden Spannungsquellen (interne und externe) bedient.

## 7.4 RUN/STOP-Betriebswahlschalter (NPLC-C100 und NPLC-M100)

Das SPS-Programm in den netPLC-Komponenten **NPLC-C100** und **NPLC-M100** kann über den Schalter in die folgenden Betriebszustände geschaltet werden:

| Position | Funktion |
|---|---|
| **RUN** | Ein geladenes SPS-Programm wird zyklisch abgearbeitet. |
| **STOP** | Die zyklische Verarbeitung des SPS-Programms wird angehalten. In diesem Zustand ist es möglich, ein neues SPS-Programm oder eine neue Firmware zu laden. |

*Tabelle 13: RUN/STOP-Betriebswahlschalter*

Wenn CoDeSys auf der Steuerung eingeloggt ist, kann von CoDeSys aus der Betriebszustand ohne Verändern der Schalterstellung gewechselt werden.

Beachten Sie Folgendes: Dies kann dazu führen, dass von CoDeSys aus die Steuerung in STOP gesetzt wird, der Schalter jedoch noch in Stellung RUN steht und die Steuerung mit dem nächsten Reset der netPLC Komponente wieder in den Betriebszustand RUN geht und dieser Wiederanlauf unbeabsichtigt ist. Stellen Sie deshalb den Schalter auf STOP, um einen unbeabsichtigten Wiederanlauf zu verhindern.

Die Position des Schalters hat Vorrang vor der Softwaresteuerung!


© Hilscher, 2009 - 2014

# 7.5 Beispiel SPS-Programm zur Batterie-Überwachung (NPLC-C100 und NPLC-M100)

Bei den netPLC-Komponenten NPLC-C100 und NPLC-M100 kann die Batteriespannung mit einer Systemfunktion StatusOfBattery überwacht werden. Informationen dazu finden Sie im Abschnitt *Batterie Status – StatusOfBattery* auf Seite 66.

Beispiel für eine Programmierung mit dieser Funktion:

```
(* a rising edge on S1.1 on the M100 evaluation board is used for triggering the measure-
ment *)
fTriggerState := %IX9.0;
IF fTriggerState AND NOT fLastTriggerState THEN
     fMeasurementEnabled := TRUE;
     iMeasurementCycles := 0;
END_IF
fLastTriggerState := fTriggerState;

IF fMeasurementEnabled THEN
     iMeasurementCycles := iMeasurementCycles + 1;
     (* check external 24 VDC power supply *)
     StatusOfExt24VDC (xDone=> fExt24VDone, xBusy=> fExt24VBusy, xError=> fExt24VError,
iError=> iExt24VError, xStatusOfExt24VDC=> fExt24VState);
     (* check 3.3 VDC power supply from PCI bus *)
     StatusOfPCIPower (xDone=> fPci3V3Done, xBusy=> fPci3V3Busy, xError=> fPci3V3Error,
iError=> iPci3V3Error, xStatusOfPciPower=> fPci3V3State);
     (* check battery voltage *)
     StatusOfBattery (xDone=> fBatteryDone, xBusy=> fBatteryBusy, xError=> fBatteryError,
iError=> iBatteryError, xStatusOfBattery=> fBatteryState, BatteryVoltage=> rBatteryVolta-
ge);
     IF fBatteryDone OR fBatteryError THEN
          fMeasurementEnabled := FALSE;
     END_IF
END_IF
```


© Hilscher, 2009 - 2014

# 8 Bus Diagnose

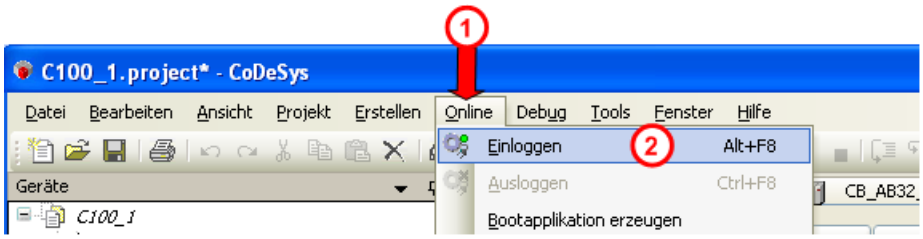
## 8.1 Bus in Betrieb nehmen und testen

**Hinweis:** Beim Laden des SPS-Programms in die Steuerung wird die Buskonfiguration immer mit übertragen.

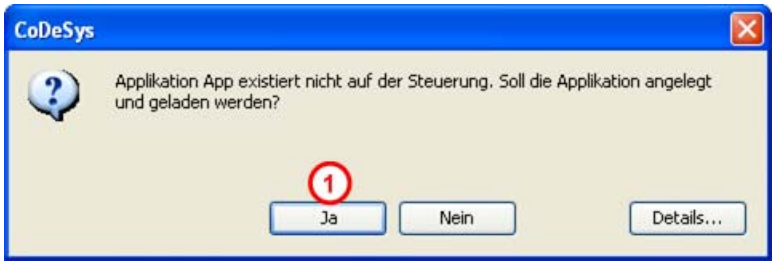
### 8.1.1 Einloggen und Programm übertragen

1. Auf der Steuerung einloggen.

➢ Wählen Sie **Online > Einloggen**.




Die Software überprüft, ob das Projekt auf der Zielbaugruppe identisch zu dem aktuell geöffneten Projekt auf dem PC ist. Befindet sich auf der Zielhardware noch kein Programm, dann erfolgt die Meldung „Applikation App existiert nicht auf der Steuerung. Soll die Applikation geladen werden?“. Sind die Projekte unterschiedlich, so erfolgt die Meldung „Warnung: Eine unbekannte Version von Applikation ‚App’ läuft gerade auf der Steuerung. Möchten Sie dennoch den aktuellen Code laden und die bestehende Applikation ersetzen?“.




2. Ladevorgang starten.

➢ Beantworten Sie die Frage „Applikation App existiert nicht auf der Steuerung. Soll die Applikation geladen werden?“ mit **Ja** (1).

Die Applikation wird auf die Steuerung geladen.

Beim Start des Ladens geht die netPLC in den STOP-Modus.


© Hilscher, 2009 - 2014

### 8.1.2 Gerätezuordnung

1. Fenster Gerätebaum öffnen.

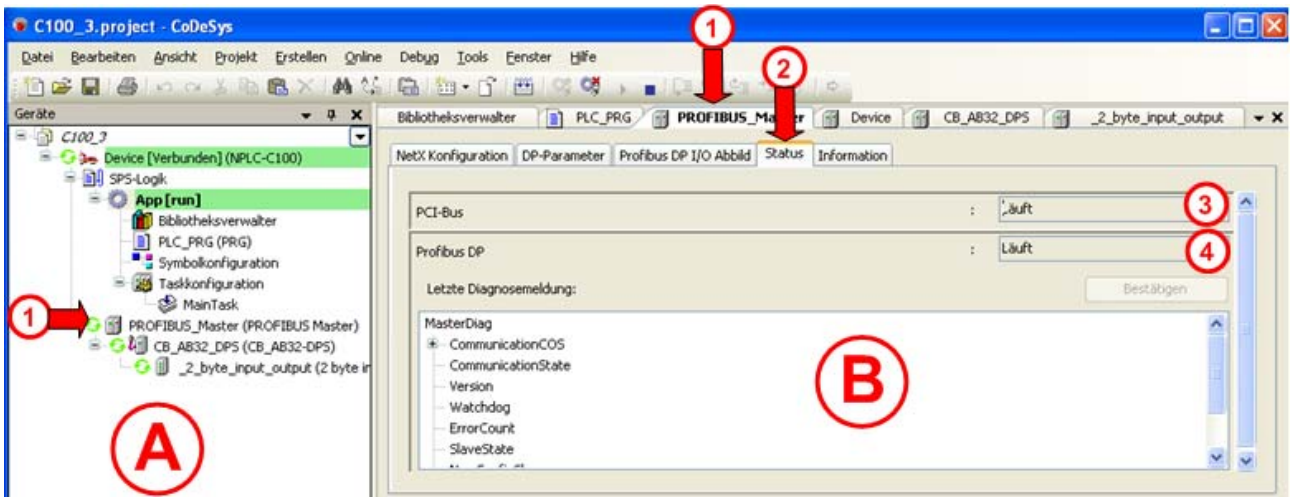



*Abbildung 48: Busdiagnose (1) Beispiel PROFIBUS*

- Öffnen Sie mit einem Doppelklick mit der linken Maus im Gerätebaum Ⓐ auf den Eintrag des Bus-Masters ① die Feldbusinformation.
- Öffnen sie mit einem Mausklick auf den Reiter **Status** ② die Businformation.
- Sind am Bussystem keine Fehler diagnostiziert worden, sehen sie obiges Bild.

  Bei ③ haben sie die Information des PCI Busses (bei einer netPLC C100 Komponente).

  Bei ④ sehen Sie die Feldbusinformation.

Im Fehlerfall sieht das Fenster wie folgt aus:

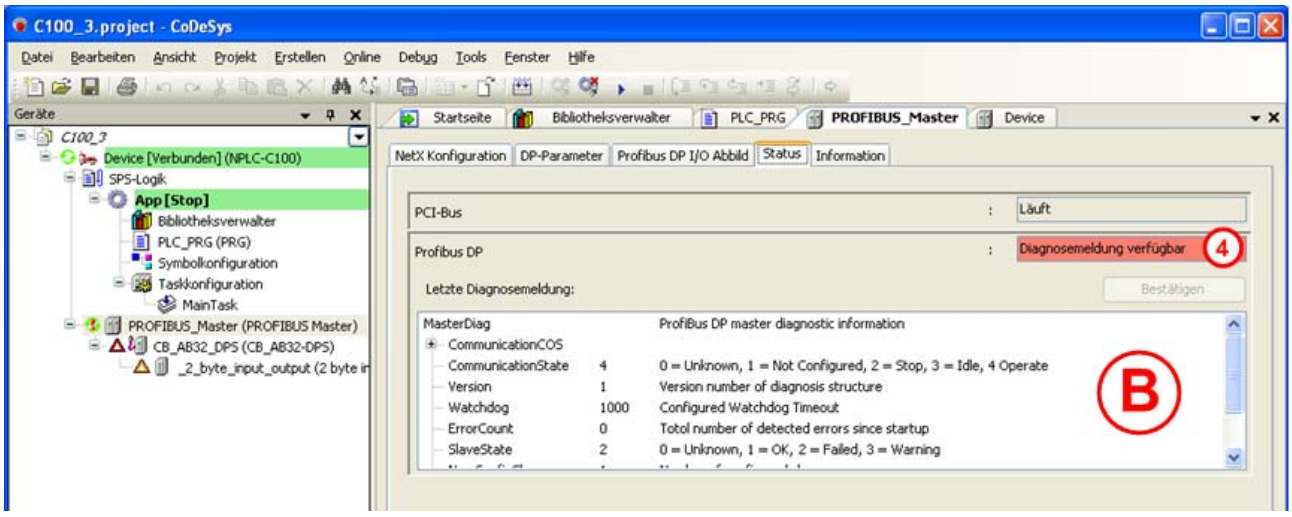



*Abbildung 49: Busdiagnose (2) Beispiel PROFIBUS*

- Bei ④ sehen Sie die rote Information, dass Diagnosemeldungen verfügbar sind. Diese werden Ihnen im Fensterbereich Ⓑ präsentiert.


© Hilscher, 2009 - 2014

### 8.1.3 Eingangs- und Ausgangsdaten prüfen

1. Mit dem I/O Abbild Monitor die Ein- und Ausgangsdaten prüfen.

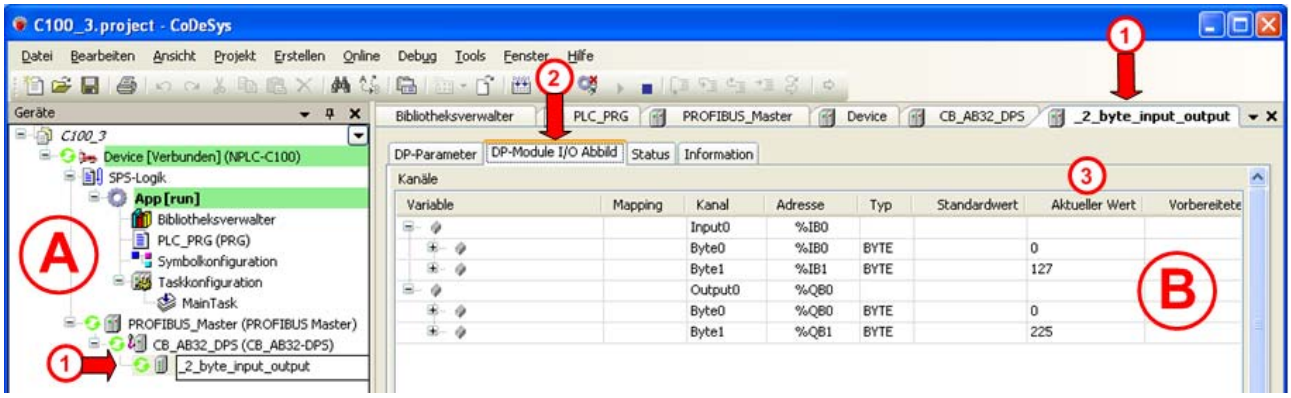



*Abbildung 50: Busdiagnose (3) Slave E/A-Daten*

➢ Wählen Sie im Gerätebaum Ⓐ den Eintrag der E/A-Daten des Slaves an ①.

➢ Wählen Sie in der zweiten Reiterleiste den Eintrag **I/O Abbild** ② an.

Im Fensterbereich Ⓑ sehen Sie jetzt die Ein- und Ausgänge des Slaves. In der Spalte **Aktueller Wert** ③ sehen Sie die aktuellen Werte der Signale.


© Hilscher, 2009 - 2014

# 9 Fehlersuche

**Allgemein**

- Prüfen Sie, ob die Voraussetzungen für den Betrieb der netPLC-Komponente erfüllt sind:

Weitere Angaben hierzu finden Sie in der Dokumentation der Hardware der jeweiligen netPLC-Komponente.

**Kabel**

- Prüfen Sie, dass die Pin-Belegung des verwendeten Kabels richtig ist.

**Konfiguration**

- Prüfen Sie, dass die Konfiguration im Master zur Konfiguration des Slaves passt. Zum Beispiel die Busadresse und Baudrate.

Nutzen Sie die Statusinformationen zu den einzelnen Gerätekomponenten des Gerätebaums des Projektes.

Nutzen sie die I/O Abbildinformationen der E/A-Signale der Slave-Komponenten des Gerätebaums.

Nutzen Sie die LED-Information an der Geräten.

## 9.1 Fehlermeldung bei der Bootapplikationserzeugung

Die folgende Fehlermeldung kann nach der Erzeugung einer Symbolkonfiguration bei der Erzeugung einer Bootapplikation erscheinen:

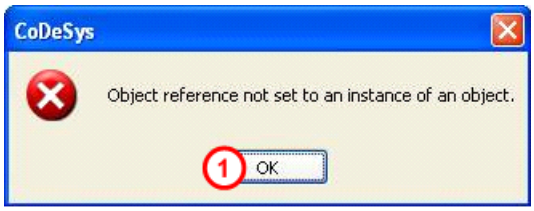



*Abbildung 51: Object reference not set…*

Gehen Sie wie folgt vor:

1. Klicken Sie **OK** (1), um das Popup zu schließen.
2. Speichern Sie das Projekt.
3. Beenden Sie die CoDeSys-Anwendung.
4. Starten Sie CoDeSys neu.
5. Laden Sie das zuvor gespeicherte Projekt.
6. Starten Sie die Bootapplikationserzeugung neu.


© Hilscher, 2009 - 2014

# 10 LEDs

## 10.1 Übersicht LEDs NPLC-C100 und NPLC-M100

Benennung und Bedeutung der LEDs der netPLC-Komponenten **NPLC-C100** und **NPLC-M100** nach Protokoll:

| LED-Benennung in der Gerätezeichnung und am Gerät | CANopen Master | DeviceNet Master | PROFIBUS DP Master |
|---|---|---|---|
| **SYS** (gelb / grün) | **SYS** *(Systemstatus)* | **SYS** *(Systemstatus)* | **SYS** *(Systemstatus)* |
| | Für Details siehe Abschnitt *LED SYS* auf Seite 76 | | |
| **APL** (gelb / grün) | **APL** *(Application, netPLC)* | **APL** *(Application, netPLC)* | **APL** *(Application, netPLC)* |
| | Für Details siehe Abschnitt *LED APL* auf Seite 77 | | |
| **COM 0** (Kommunikationsstatus) (rot / grün) | **-** *(Kommunikationsstatus)* | **MNS** *(Kommunikationsstatus)* Für Details siehe Abschnitt *LED DeviceNet Master* auf Seite 77 | **COM** *(Kommunikationsstatus)* Für Details siehe Abschnitt *LED PROFIBUS DP-Master* auf Seite 78 |
| **COM 1** (Kommunikationsstatus) (rot / grün) | (*nicht verwendet*) | (*nicht verwendet*) | (*nicht verwendet*) |

*Tabelle 14: Übersicht LEDs NPLC-C100 und NPLC-M100 und deren Bedeutung nach Protokoll (Feldbus-System)*

## 10.2 Übersicht LEDs NHMI J500Q5-DP-DP/DPM/CDS/QV (PROFIBUS DP Master)

Benennung und Bedeutung der LEDs der netPLC-Komponente netHMI PLC **NHMI J500Q5-DP-DP/DPM/CDS/QV**:

| LED | Farbe | Bedeutung | Details siehe |
|---|---|---|---|
| SYS | Duo LED gelb/grün | Systemstatus | Abschnitt *LED SYS* auf Seite 76 |
| COM | Duo LED rot/grün | Kommunikations-status PROFIBUS DP | Abschnitt *LED PROFIBUS DP-Master* auf Seite 78 |
| - | - | - | - |
| APL | Duo LED gelb/grün | Applikationsstatus (SPS) | Abschnitt *LED APL* auf Seite 77 |

*Tabelle 15: Übersicht LEDs NHMI J500Q5-DP-DP/DPM/CDS/QV*


© Hilscher, 2009 - 2014

## 10.3 Übersicht LEDs NHMI J500Q5-RE-DP/PNM/CDS/QV (PROFINET IO Controller)

Benennung und Bedeutung der LEDs der netPLC-Komponente netHMI PLC **NHMI J500Q5-RE-DP/PNM/CDS/QV**:

| LED | Farbe | Bedeutung | Details siehe |
|---|---|---|---|
| **SYS** | Duo LED gelb/grün | Systemstatus | Abschnitt *LED SYS* auf Seite 76 |
| **SF** | Duo LED rot/grün | Systemfehler | Abschnitt *LEDs PROFINET IO Controller* auf Seite 78 |
| **BF** | Duo LED rot/grün | Busfehler | |
| **APL** | Duo LED gelb/grün | Applikationsstatus (SPS) | Abschnitt *LED APL* auf Seite 77 |

*Tabelle 16: Übersicht LEDs NHMI J500Q5-RE-DP/PNM/CDS/QV*

## 10.4 LED SYS

In der nachfolgenden Tabelle ist die Bedeutung der System-LED der netPLC-Komponente beschrieben.

| LED | Farbe | Zustand | Bedeutung |
|---|---|---|---|
| **SYS** | **Duo LED gelb/grün** | | |
| | (grün) | Ein | Betriebssystem läuft |
| | (grün/gelb) | Blinkend gelb/grün | Bootloader wartet auf Firmware. |
| | (gelb) | Blinkend | Schwerer Systemfehler. SPS in Stop. |
| | (gelb) | Ein | Bootloader wartet auf Software. |
| | /gelb/grün/ aus) | Blinkend fortlaufend | SecMem nicht lesbar. |
| | | Aus | Versorgungsspannung für das Gerät fehlt oder Hardware-defekt. |

*Tabelle 17: System-LED*


© Hilscher, 2009 - 2014

## 10.5 LED APL

| LED | Farbe | Zustand | Bedeutung |
|---|---|---|---|
| **APL** | **Duo LED grün/gelb** | | |
| | (grün) | Ein | SPS im Zustand RUN.<br>SPS-Programm wird ausgeführt. |
| | (grün) | Zyklisch blinkend 1 Hz | Kein bootbares SPS-Programm vorhanden. |
| | (gelb) | Ein | SPS im Zustand STOP.<br>SPS-Programm wird nicht ausgeführt. |
| | (gelb) | Zyklisch blinkend 4 Hz | Unerwarteter Fehler bei der Ausführung des SPS-Programms aufgetreten. |

*Tabelle 18: LED APL CoDeSys*

## 10.6 LED CANopen Master

Für das CANopen-Master Protokoll gibt es kein LED-Signal auf den netPLC-Komponenten.

## 10.7 LED DeviceNet Master

In der nachfolgenden Tabelle ist die Bedeutung der LED beschrieben, wenn die Firmware des DeviceNet-Master-Protokolls in das Gerät geladen wurde.

| LED | Farbe | Zustand | Bedeutung |
|---|---|---|---|
| **MNS**<br>Benennung in der Geräte-zeichnung: COM0 | **Duo-LED rot/grün** | | |
| | (grün) | Ein | Gerät ist online und hat eine oder mehrere Verbindungen aufgebaut |
| | (grün) | Blinkt | Gerät ist online und hat keine Verbindung aufgebaut |
| | (grün/rot/ aus) | Grün/Rot/ Aus | Selbsttest nach Spannung einschalten:<br>Grün ein für 0,25 s, dann rot ein für 0,25 s, dann aus |
| | (rot) | Ein | Kritischer Verbindungsfehler; Gerät hat einen Netzwerkfehler erkannt: doppelte MAC-ID oder schwerer Fehler im CAN-Netzwerk (CAN-Bus-Off) |
| | (rot) | Blinkt | Verbindungsüberwachungszeit abgelaufen |
| | (aus) | Aus | Nach Start des Gerätes und während der Prüfung auf doppelte MAC-ID |

*Tabelle 19: LEDs DeviceNet-Master*


© Hilscher, 2009 - 2014

## 10.8 LED PROFIBUS DP-Master

Benennung in der Gerätezeichnung: COM0

In der nachfolgenden Tabelle ist die Bedeutung der LED beschrieben, wenn die Firmware des PROFIBUS DP-Master-Protokolls in das Gerät geladen wurde.

| LED | Farbe | Zustand | Bedeutung |
|---|---|---|---|
| **COM** | **Duo-LED rot/grün** | | |
| | (grün) | Azyklisch blinkend | Keine Konfiguration oder Stack-Fehler |
| | (grün) | Zyklisch blinkend | Profibus ist konfiguriert, aber Buskommunikation noch nicht freigegeben von der Application |
| | (grün) | Ein | Kommunikation zu allen Slaves hergestellt |
| | (rot) | Zyklisch blinkend | Kommunikation zu mindestens einem Slave unterbrochen |
| | (rot) | Ein | Kommunikation zu allen/einem Slave unterbrochen oder es ist ein anderer schwerwiegender Fehler aufgetreten.<br>Im Redundanten Mode: Der aktive Master wurde nicht gefunden. |

*Tabelle 18: LEDs PROFIBUS DP-Master*

## 10.9 LEDs PROFINET IO Controller

| LED | Farbe | Zustand | Bedeutung |
|---|---|---|---|
| **SF** | **Duo-LED rot/grün** | | |
| | (rot) | Ein | *(*zusammen *mit BF „rot Ein“)*<br>**Keine gültige Master-Lizenz** |
| | (rot) | Blinkt zyklisch mit 2 Hz | **Systemfehler**: ungültige Konfiguration, Überwachungsfehler oder interner Fehler |
| | (aus) | Aus | Kein Fehler |
| **BF** | **Duo-LED rot/grün** | | |
| | (rot) | Ein | **Keine Verbindung**: Kein Link.<br>*oder (*zusammen *mit SF „rot Ein“)*<br>Keine gültige Master-Lizenz |
| | (rot) | Blinkt zyklisch mit 2 Hz | **Konfigurationsfehler**: Nicht alle konfigurierten IO-Devices sind verbunden. |
| | (aus) | Aus | Kein Fehler |

*Tabelle 20: LEDs PROFINET IO-RT-Controller*


© Hilscher, 2009 - 2014

# 11 Fehlernummern

## 11.1 Fehlernummern Funktionen HilPlcSysFunctions

Die Fehlernummern haben den Datentyp INT.

| Fehlernummer | Name<br>Beschreibung |
|---|---|
| 0 | NO_ERROR<br>Kein Fehler aufgetreten |
| 1000 | INTERNAL_ERROR<br>Unerwarteter interner Fehler aufgetreten |
| 1001 | BATTERY_BAD_ERROR<br>Der Status der Batterie ist “Schlecht”. Die Batterie muss gewechselt werden. |
| 1010 | FCT_NOT_SUPPORTED_ERROR<br>Diese Funktion wird nicht unterstützt. |

*Tabelle 21: Fehlernummern Funktionen HilPlcSysFunctions_netPLC*


© Hilscher, 2009 - 2014

# 12 Technische Daten der Protokolle

## 12.1 CANopen Master und DeviceNet Master

Für CANopen Master und DeviceNet Master wird dieselbe Firmware-Datei verwendet. Die CANopen-Kommunikationsfunktionen laufen (als IEC-Code) in CoDeSys Control.

### CANopen Master

| Parameter | Beschreibung |
|---|---|
| Maximale Anzahl CANopen Knoten | 126 |
| Maximale Anzahl zyklischer Eingangsdaten | 3584 Bytes<br>Bei NPLC-C100 können davon max. 512 Bytes für Host-Eingangsdaten verwendet werden. |
| Maximale Anzahl zyklischer Ausgangsdaten | 3584 Bytes<br>Bei NPLC-C100 können davon max. 512 Bytes für Host-Ausgangsdaten verwendet werden. |
| Maximale Anzahl empfangener PDOs | 512 |
| Maximale Anzahl übertragener PDOs | 512 |
| Austausch von Prozessdaten | Via PDO-Transfer:<br>- synchronisiert,<br>- fernabgefragt und<br>- event-gesteuert (Datenänderung) |
| Funktionen | Emergency-Message (Consumer und Producer)<br>Node-Guarding / Life-Guarding, Heartbeat<br>PDO-Mapping<br>NMT-Master<br>SYNC-Protokoll (Producer) |
| Baudrate | 10 kBits/s,<br>20 kBits/s,<br>50 kBits/s,<br>100 kBits/s,<br>125 kBits/s,<br>250 kBits/s,<br>500 kBits/s,<br>800 kBits/s,<br>1 MBits/s |
| Daten-Transport-Layer | CAN-Frames |
| CAN-Frame-Typ für CANopen | 11 Bit |
| Bezug auf Firmware/Stack Version | V4.0.x.x |

*Tabelle 22: Technische Daten CANopen-Master Protokoll*


© Hilscher, 2009 - 2014

### DeviceNet Master

| Parameter | Beschreibung |
|---|---|
| Maximale Anzahl DeviceNet Slaves | 63 |
| Maximale Anzahl aller zyklischer Eingangsdaten | 3584 Bytes<br>Bei NPLC-C100 können davon max. 512 Bytes für Host-Eingangsdaten verwendet werden. |
| Maximale Anzahl aller zyklischer Ausgangsdaten | 3584 Bytes<br>Bei NPLC-C100 können davon max. 512 Bytes für Host-Ausgangsdaten verwendet werden. |
| Maximale Anzahl zyklischer Eingangsdaten | 255 Bytes/Verbindung |
| Maximale Anzahl zyklischer Ausgangsdaten | 255 Bytes/Verbindung |
| Maximaler Umfang Konfigurationsdaten | 1000 Bytes/Slave |
| Verbindungen | Bit-Strobe<br>Change of State<br>Cyclic<br>Poll<br>Explicit Peer-to-Peer Messaging (nur zur Parametrierung) |
| Funktionen | Quick Connect |
| Fragmentation | Explicit und E/A |
| UCMM | Unterstützt |
| Objekte | Identity Object (Class Code 0x01)<br>Message Router Object (Class Code 0x02)<br>DeviceNet Object (Class Code 0x03)<br>Connection Object (Class Code 0x05)<br>Acknowledge Handler Object (Class Code 0x06) |
| Baudrate | 125 kBits/s,<br>250 kBit/s,<br>500 kBit/s<br>Automatische Baudratenerkennung wird nicht unterstützt |
| Daten-Transport-Layer | CAN Frames |
| Einschränkungen | Nutzdatenübertragung durch netPLC nur über ‘IO Connections’ |
| Bezug auf Firmware/Stack Version | V4.0.x.x/2.3.x.x |

*Tabelle 23: Technische Daten DeviceNet-Master Protokoll*


© Hilscher, 2009 - 2014

# 12.2 PROFIBUS DP-Master

## 12.2.1 NPLC-C100-DP und NPLC M100-DP

Die technischen Daten in der nachfolgenden Tabelle gelten für die Gerätetypen **NPLC-C100-DP** und **NPLC M100-DP**:

| Parameter | Beschreibung |
|---|---|
| Maximale Anzahl PROFIBUS DP-Slaves | 125 (DPV0/DPV1) |
| Maximale Anzahl aller zyklischer Eingangsdaten | 3584 Bytes<br>Bei NPLC-C100 können davon max. 512 Bytes für Host-Eingangsdaten verwendet werden. |
| Maximale Anzahl aller zyklischer Ausgangsdaten | 3584 Bytes<br>Bei NPLC-C100 können davon max. 512 Bytes für Host-Ausgangsdaten verwendet werden. |
| Maximale Anzahl zyklischer Eingangsdaten | 244 Bytes pro Slave |
| Maximale Anzahl zyklischer Ausgangsdaten | 244 Bytes pro Slave |
| Konfigurationsdaten | Max. 244 Bytes pro Slave |
| Parametrierungsdaten pro Slave | 7 Bytes Standardparameter pro Slave<br>Max. 237 Bytes pro Slave applikations-spezifische Parameter |
| Azyklische Kommunikation | Mit der CoDeSys-Bibliothek *IIoDrvDPV1C1* können folgende DPV1-Klasse 1-Dienste genutzt werden:<br>- Lesen, Schreiben<br>- Alarm und AlarmAck |
| Maximale Anzahl azyklischer Daten (read/write) | 240 Bytes pro Slave und Telegramm |
| Baudrate | 9,6 kBits/s,<br>19,2 kBits/s,<br>31,25 kBits/s,<br>45,45 kBits/s,<br>93,75 kBits/s,<br>187,5 kBits/s,<br>500 kBits/s,<br>1, 5 MBits/s,<br>3 MBits/s,<br>6 MBits/s,<br>12 MBit/s<br>Automatische Baudratenerkennung wird nicht unterstützt. |
| Datentransport-Layer | PROFIBUS FDL |
| Einschränkungen | DPV1-Dienste der Klasse 2 werden nicht unterstützt.<br>DPV2-Dienste sind nicht implementiert.<br>Redundanzfunktion wird nicht unterstützt.<br>Configuration in Run (CiR) wird nicht unterstützt. |
| Bezug auf Firmware/Stack Version | Firmware V4.0 (PROFIBUS DP-Master V2.6) |

*Tabelle 24: Technische Daten PROFIBUS DP-Master Protokoll*


© Hilscher, 2009 - 2014

### 12.2.2 NHMI J500Q5-DP-DP

Die technischen Daten in der nachfolgenden Tabelle gelten für den Gerätetyp **NHMI J500Q5-DP-DP**:

| Parameter | Beschreibung |
|---|---|
| Maximale Anzahl PROFIBUS DP-Slaves | 31 (DPV0/DPV1) |
| Maximale Anzahl aller zyklischer Eingangsdaten | 512 Bytes |
| Maximale Anzahl aller zyklischer Ausgangsdaten | 512 Bytes |
| Maximale Anzahl zyklischer Eingangsdaten | 244 Bytes pro Slave |
| Maximale Anzahl zyklischer Ausgangsdaten | 244 Bytes pro Slave |
| Konfigurationsdaten | Max. 244 Bytes pro Slave |
| Parametrierungsdaten pro Slave | 7 Bytes Standardparameter pro Slave<br>Max. 237 Bytes pro Slave applikations-spezifische Parameter |
| Azyklische Kommunikation | Mit der CoDeSys-Bibliothek IIoDrvDPV1C1 können folgende DPV1-Klasse 1-Dienste genutzt werden:<br>- Lesen, Schreiben<br>- Alarm und AlarmAck |
| Maximale Anzahl azyklischer Daten (read/write) | 240 Bytes pro Slave und Telegramm |
| Baudrate | 9,6 kBits/s,<br>19,2 kBits/s,<br>31,25 kBits/s,<br>45,45 kBits/s<br>93,75 kBits/s,<br>187,5 kBits/s,<br>500 kBits/s,<br>1, 5 MBits/s,<br>3 MBits/s,<br>6 MBits/s,<br>12 MBit/s<br>Automatische Baudratenerkennung wird nicht unterstützt. |
| Daten-Transport-Layer | PROFIBUS FDL |
| Einschränkungen | DPV1-Dienste der Klasse 2 werden nicht unterstützt.<br>DPV2-Dienste werden nicht unterstützt.<br>Redundanzfunktion wird nicht unterstützt.<br>Configuration in Run (CiR) wird nicht unterstützt. |
| Bezug auf Firmware/Stack Version | Firmware V4.0 (PROFIBUS DP-Master V2.6) |

*Tabelle 25: Technische Daten PROFIBUS DP-Master Protokoll*


© Hilscher, 2009 - 2014

## 12.3 PROFINET IO-RT-Controller

| Parameter | Beschreibung |
|---|---|
| Maximale Anzahl PROFINET IO-Devices | 31 |
| Maximale Gesamtanzahl zyklischer Eingangsdaten | 512 Bytes (inklusive IOxS-Statusbytes) |
| Maximale Gesamtanzahl zyklischer Ausgangsdaten | 512 Bytes (inklusive IOxS-Statusbytes) |
| Maximale Anzahl zyklischer Eingangsdaten | 512 Bytes pro IO-Device (= IOCR-Datenlänge inklusive IOxS-Statusbytes) |
| Maximale Anzahl zyklischer Ausgangsdaten | 512 Bytes pro IO-Device (= IOCR-Datenlänge inklusive IOxS-Statusbytes) |
| Azyklische Kommunikation | Mit der CoDeSys-Bibliothek *IIoDrvPNAcyclicServices* können folgende Dienste genutzt werden:<br>Read Record, Write Record<br>Maximal 1392 Bytes pro Telegramm und Request |
| Context-Management durch CL-RPC | Unterstützt |
| Minimale Zykluszeit | 1 ms<br>IO-Devices können mit unterschiedlichen Zykluszeiten konfiguriert werden. |
| Baudrate | 100 MBit/s<br>Voll-Duplex |
| Datentransport-Layer | Ethernet II, IEEE 802.3 |
| Konfigurationsdatei | Maximal 1 MByte |
| Einschränkungen | Keine Alarmbehandlung<br>‚Fast Startup' von PROFINET IO-Device(s) wird nicht unterstützt.<br>RT über UDP wird nicht unterstützt.<br>Multicast Kommunikation wird nicht unterstützt.<br>DHCP nicht unterstützt (weder für PROFINET IO-Controller noch für PROFINET IO-Devices)<br>Eine IOCR pro IO-Device<br>Der NameOfStation des IO-Controller kann nicht mit dem Dienst 'DCP SET NameOfStation' gesetzt werden, sondern nur durch Konfiguration des IO-Controllers.<br>Der Puffer für die Diagnose Daten eines IO-Devices wird im Falle mehrerer Diagnoseereignisse überschrieben. Nur ein (das letzte) Diagnoseereignis wird zu einem Zeitpunkt gespeichert. Wenn ein Diagnoseereignis mehr als 200 Bytes Diagnosedaten erzeugt, dann werden nur die ersten 200 Bytes gespeichert.<br>Die verwendbare (kleinste) Zykluszeit ist abhängig von der Anzahl der IO-Devices, der Anzahl verwendeter Eingangs- und Ausgangsdaten. Die Zykluszeit, die Anzahl konfigurierter IO-Devices und die Anzahl der E/A-Daten hängen voneinander ab. Es ist aus Performencegründen z. B. nicht möglich 31 IO-Devices mit einer Zykluszeit von 1 ms zu betreiben.<br>Die Größe der Buskonfigurationsdatei ist durch die Größe der RAM Disk begrenzt (1 MByte)<br>Der Dienst WriteMultiple-Record wird nicht unterstützt |
| Bezug auf Firmware/Stack Version | Firmware V4.0 (PROFINET IO-Controller V2.6) |

*Tabelle 26: Technische Daten PROFINET IO-Controller Protokoll*


© Hilscher, 2009 - 2014

# 13 Anhang

## 13.1 netPLC CoDeSys Server Programm

Das netPLC CoDeSys Server Programm stellt eine Verbindung zwischen dem 3S Gateway (und damit zum CoDeSys Programmiersystem) auf der einen Seite und über den Gerätetreiber (cifX Device Driver) auf der anderen Seite zur netPLC C100 her. Das 3S Gateway kann auf demselben PC oder auf einem anderen PC installiert sein (verbunden über das lokale Ethernet Netzwerk).

### 13.1.1 netPLC CoDeSys Server starten

Der netPLC CoDeSys Server wird mit der Installation in den Autostart installiert, so dass der Server mit jedem Start des PC automatisch gestartet wird.

Um den Server manuell zu starten, wählen Sie **Start > Alle Programme > Hilscher GmbH > netPLC CoDeSys Server**.

Wenn das netPLC CoDeSys Server Programm gestartet ist, dann erscheint im Systemtray das Icon

*Abbildung 52: netPLC CoDeSys Server Tray Icon*


© Hilscher, 2009 - 2014

### 13.1.2 netPLC CoDeSys Server Fenster

Mit einem Doppelklick auf das Tray Icon öffnet sich das Fenster des netPLC CoDeSys Servers.

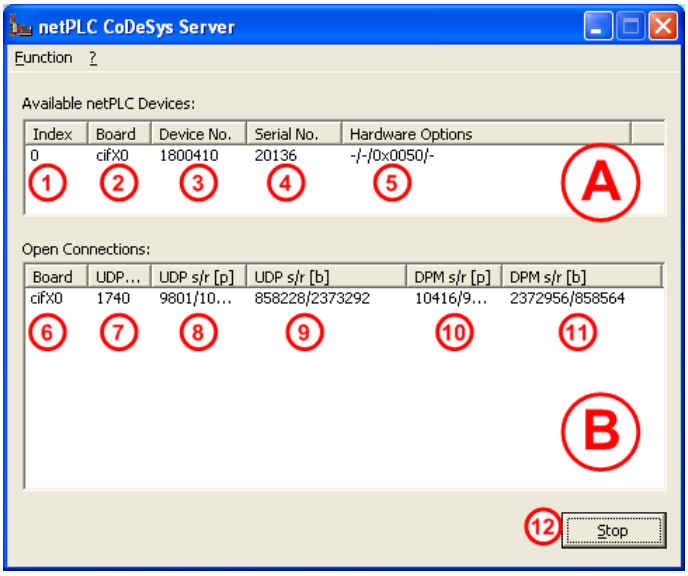



*Abbildung 53: netPLC CoDeSys Server Fenster*

Die angezeigten Elemente haben folgende Bedeutung:

Im Fensterbereich Ⓐ werden die verfügbaren Komponenten angezeigt.

Im Fensterbereich Ⓑ werden die Komponenten angezeigt, zu denen eine Verbindung hergestellt wurde.

| Element | Beschreibung |
|---|---|
| ① Index | Laufende Nummer |
| ② Board | Zeigt zu welcher Slot-SPS NPLC-C100 (cifX0, cifX1, …) der CoDeSys Server eine Verbindung aufbauen kann. |
| ③ Device No. | Artikelnummer der Komponente |
| ④ Serial No. | Fertigungsnummer der Komponente |
| ⑤ Hardware Options | Schnittstellenidentifikationskennung |
| ⑥ Board | Zeigt an, auf welche Slot-SPS NPLC-C100 (cifX0, cifX1, …) der Zugriff über den PC-Karten Gerätetreiber hergestellt ist. |
| ⑦ UDP Port | Port, über den der netPLC CoDeSys Server ‘Pakete’ vom 3S Gateway empfängt bzw. sendet. |
| ⑧ UDP s/r [p] | Zähler für gesendete Pakete (UDP s = senden) sowie für empfangene Pakete (UDP r = empfangen) über die Verbindung zum 3S Gateway. |
| ⑨ UDP s/r [b] | Zähler für gesendete Bytes (UDP s = senden) sowie für empfangene Bytes (UDP r = empfangen) über die Verbindung zum 3S Gateway. |
| ⑩ DPM s/r [p] | Zähler für gesendete Pakete (UDP s = senden) sowie für empfangene Pakete (UDP r = empfangen), d. h. die Verbindung über den Gerätetreiber zur NPLC-C100. |
| ⑪ DPM s/r [b] | Zähler für gesendete Bytes (UDP s = senden) sowie für empfangene Bytes (UDP r = empfangen), d. h. die Verbindung über den Gerätetreiber zur NPLC-C100. |

*Tabelle 27: netPLC CoDeSys Server Fenster – Elemente*


© Hilscher, 2009 - 2014

### 13.1.3 netPLC CoDeSys Server beenden

Zum Beenden des netPLC CoDeSys Servers mit der rechten Maustaste auf das Tray Icon klicken. Aus dem Kontextmenü **Stop** wählen oder die Schaltfläche **Stop** (12) im Fenster des Servers klicken.

Die Sicherheitsabfrage mit **Ja** beantworten, um den netPLC CoDeSys Server zu beenden.


© Hilscher, 2009 - 2014

# 13.2 Beispielprojekte

## 13.2.1 netPLC CoDeSys Produkt-DVD

Auf der netPLC CoDeSys Produkt-DVD sind mehrere Beispielprojekte vorhanden. Eine kurze Beschreibung folgt in den nächsten Abschnitten.

Zum Öffnen, Laden und Starten des jeweiligen SPS-Programms sind grundsätzlich folgende Schritte notwendig:

- Verwenden Sie in CoDeSys das Menü **Datei > Projekt öffnen**, um das entsprechende Projekt zu öffnen.
- Setzen Sie den Aktiven Pfad.
- Verwenden Sie in CoDeSys das Menü **Online > Einloggen auf App [Device: SPS-Logik]**, um das Programm an die netPLC C100 zu übertragen (oder die Tastenkombination Alt+F8).
- Verwenden Sie in CoDeSys das Menü **Online > Bootprojekt erzeugen für App [Device: SPS-Logik]**.
- Starten Sie das SPS-Programm mit **Online > Start App [Device: SPS-Logik]** (oder mit der Tastenkombination F5).

### 13.2.1.1 netPLC System Function

| | |
|---|---|
| **Projektname** | netPLC System Functions.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\netPLC System Functions |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Gerät eine netPLC NPLC-C100-DP. |
| **SPS-Programm** | Das SPS-Programm zeigt die Verwendung der Systemfunktionen<br>StatusOfBattery,<br>StatusOfExt24V und<br>StatusOfPCIPower |

*Tabelle 28: Beispiel – netPLC System Function*

Die Systemfunktionen sind in Abschnitt *Funktionsbibliothek erweitern* ab Seite 63 beschrieben.

### 13.2.1.2 netPLC Open Process Control (OPC)

| | |
|---|---|
| **Projektname** | Open Process Control.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\<br>netPLC Open Process Control (OPC) |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Gerät eine netPLC NPLC-C100-DP. |
| **SPS-Programm** | Das SPS-Programm zeigt, wie Variablen angelegt werden und diese mittels der Symbolkonfiguration für den OPC Server zugänglich gemacht werden.<br>Von einem OPC-Klient kann dann auf die Variable opc_qb0 ein Wert (0 ... 255) geschrieben werden. Das SPS-Programm überträgt den Wert der Variablen opc_qb0 auf die Variable opc_ib0. Der OPC-Klient kann diesen Wert dann aus opc_ib0 lesen.<br>Das SPS-Programm führt den gleichen Ablauf mit der Variablen opc_qb1 und opc_ib1 aus. |

*Tabelle 29: Beispiel – netPLC Open Process Control*


© Hilscher, 2009 - 2014

Um das Projekt zu verwenden, gehen Sie wie folgt vor:

- Verwenden Sie in CoDeSys das Menü **Datei > Projekt öffnen**, um das Projekt zu öffnen.
- Setzen Sie den Aktiven Pfad.
- Merken Sie sich die Adressangabe, wenn der aktive Pfad gesetzt wurde. Diese Adresse muss beim OPC Konfigurator wieder eingegeben werden.
- Verwenden Sie in CoDeSys das Menü **Online > Einloggen auf App [Device: SPS-Logik]**, um das Programm an die netPLC C100 zu übertragen (oder die Tastenkombination Alt+F8).
- Verwenden Sie in CoDeSys das Menü **Online > Bootprojekt erzeugen für App [Device: SPS-Logik]**.
- Starten Sie das SPS-Programm mit **Online > Start App [Device: SPS-Logik]** (oder mit der Tastenkombination F5).
- Erstellen Sie mit dem OPC Konfigurator die OPC Konfiguration.
  Dies ist ausführlich im Abschnitt *OPC Konfigurator* ab Seite 57 beschrieben.
- Stellen Sie vom OPC-Klient eine Verbindung zum OPC Server her.
- Browsen Sie die Items (Item anzeigen lassen).
- Wählen Sie die Items opc_qb0, opc_qb1, opc_ib0 und opc_ib1 aus.
- Schreiben Sie Werte zwischen 0 und 255 auf die Items opc_qb0 und opc_qb1 und prüfen die gelesenen Werte für die Items opc_ib0 bzw. opc_ib1.

#### 13.2.1.3 netPLC Real Time Clock

| | |
|---|---|
| **Projektname** | netPLC Real Time Clock.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\netPLC Real Time Clock |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Gerät eine netPLC NPLC-C100-DP. |
| **SPS-Programm** | Das SPS-Programm zeigt, wie mittels SetDateAndTime und GetDateAndTime Datum und Uhrzeit auf der netPLC C100 eingestellt bzw. ausgelesen wird. |

*Tabelle 30: Beispiel – netPLC Real Time Clock*

#### 13.2.1.4 NPLC-C100-DP PROFIBUS Simple IO

| | |
|---|---|
| **Projektname** | NPLC-C100-DP Profibus Simple IO.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\netPLC PROFIBUS Simple IO |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Geräte eine netPLC NPLC-C100-DP als PROFIBUS-DP Master und ein PROFIBUS-DP Slave Gerät CBAB32-DPS. Das PROFIBUS-DP Slave Gerät hat zwei Byte Eingangsdaten (%IB0 und %IB1) und zwei Byte Ausgangsdaten (%QB0 und %QB1). |
| **SPS-Programm** | Das SPS-Programm hat auf %QB0 einen Zähler, der mit jedem SPS-Zyklus um 1 erhöht wird. Das SPS-Programm kopiert den Inhalt des Eingangsbyte %IB1 auf das Ausgangsbyte %QB1. |

*Tabelle 31: Beispiel – NPLC-C100-DP Profibus Simple IO*


© Hilscher, 2009 - 2014

#### 13.2.1.5 NPLC-C100-DP with Host IOs

| | |
|---|---|
| **Projektname** | NPLC-C100 Host IOs.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\NPLC-C100-DP with Host IOs |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Geräte eine netPLC NPLC-C100-DP als PROFIBUS-DP Master und ein PROFIBUS-DP Slave Gerät CBAB32-DPS. Das PROFIBUS-DP Slave Gerät hat zwei Byte Eingangsdaten (%IB0 und %IB1) und zwei Byte Ausgangsdaten (%QB0 und %QB1). |
| **SPS-Programm** | Das SPS-Programm hat auf %QB0 einen Zähler, der mit jedem SPS-Zyklus um 1 erhöht wird. Das SPS-Programm kopiert den Inhalt des Eingangsbyte %IB1 auf das Ausgangsbyte %QB1. Die Host E/A-Daten liegen ab %IB2 und ab %QB2. %QB2 wird jedem SPS-Zyklus um 1 erhöht. %IB513 wird auf %QB513 kopiert. Die Summe der Werte %IB5 und %IB0 wird in %QB5 gespeichert. |

*Tabelle 32: Beispiel – NPLC-C100-DP with Host IOs*

#### 13.2.1.6 NPLC-C100-DP DPV1

| | |
|---|---|
| **Projektname** | NPLC-C100 DPV1.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\NPLC-C100-DP DPV1 |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Geräte eine netPLC NPLC-C100-DP als PROFIBUS-DP Master und ein PROFIBUS-DP Slave Gerät Wago 750-333. Das PROFIBUS-DP Slave Gerät hat ein Eingangsdatenbyte (%IB0) und ein Ausgangsdatenbyte (%QB0). |
| **SPS-Programm** | Das Beispiel zeigt ein DPV1 Klasse 1 Lesedienst auf Slot 0 Index 8. |

*Tabelle 33: Beispiel – NPLC-C100-DP DPV1*

### 13.2.2 netHMI PLC Produkt-DVD

Auf der netHMI PLC Produkt-DVD befindet sich ein Beispielprojekt für das Gerät **NHMI J500Q5-DP-DP/DPM/CDS/QV** (PROFIBUS DP Master).

Zum Öffnen, Laden und Starten des SPS-Programms sind folgende Schritte notwendig:

- Verwenden Sie in CoDeSys das Menü **Datei > Projekt öffnen**, um das entsprechende Projekt zu öffnen.
- Setzen Sie den Aktiven Pfad.
- Verwenden Sie in CoDeSys das Menü **Online > Einloggen auf App [Device: SPS-Logik]**, um das Programm an das netHMI PLC-Gerät zu übertragen (oder die Tastenkombination Alt+F8).
- Verwenden Sie in CoDeSys das Menü **Online > Bootprojekt erzeugen für App [Device: SPS-Logik]**.
- Starten Sie das SPS-Programm mit **Online > Start App [Device: SPS-Logik]** (oder mit der Tastenkombination F5).

**NHMI J500Q5-Project**

| | |
|---|---|
| **Projektname** | NHMI J500Q5.project |
| **Pfad** | Supplements & Examples\CoDeSys Sample Projects\NHMI J500Q5.project |
| **Kurzbeschreibung** | Das Projekt beinhaltet als Gerät ein netHMI PLC **NHMI J500Q5-DP-DP/DPM** |
| **SPS-Programm** | Das SPS-Programm zeigt die Verwendung eines einfachen Counters |

*Tabelle 34: Beispiel – NHMI J500Q5*


© Hilscher, 2009 - 2014

## 13.3 Auslastungsinformationen

Informationen über den benötigten und freien Speicherbereich bekommen Sie beim Erstellen des Codes für die netPLC-Komponente (Erstellen -> Code erzeugen) im Meldungsbereich. (Scrollen Sie den Meldungsbereich nach unten.)

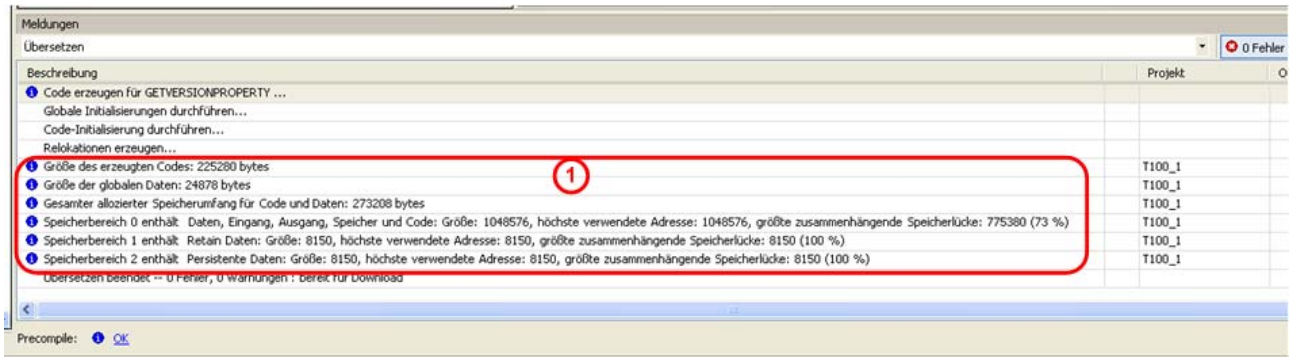
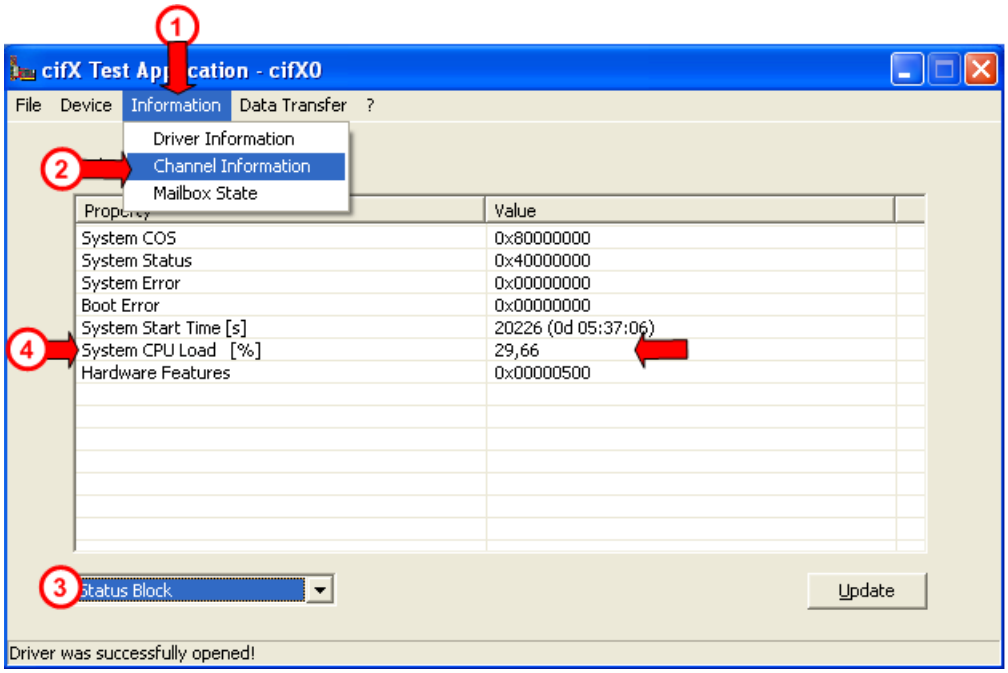

Meldungen
Übersetzen
0 Fehler
Beschreibung
Projekt
Code erzeugen für GETVERSIONPROPERTY ...
Globale Initialisierungen durchführen...
Code-Initialisierung durchführen...
Relokationen erzeugen...
Größe des erzeugten Codes: 225280 bytes
Größe der globalen Daten: 24878 bytes
Gesamter allozierter Speicherumfang für Code und Daten: 273208 bytes
Speicherbereich 0 enthält Daten, Eingang, Ausgang, Speicher und Code: Größe: 1048576, höchste verwendete Adresse: 1048576, größte zusammenhängende Speicherlücke: 775380 (73 %)
Speicherbereich 1 enthält Retain Daten: Größe: 8150, höchste verwendete Adresse: 8150, größte zusammenhängende Speicherlücke: 8150 (100 %)
Speicherbereich 2 enthält Persistente Daten: Größe: 8150, höchste verwendete Adresse: 8150, größte zusammenhängende Speicherlücke: 8150 (100 %)
T100_1
Precompile:
OK

*Abbildung 54: Speicherplatzbedarf des Projektes*

Die Speicherbedarfsinformationen finden Sie in der obigen Abbildung im Bereich ①.

Informationen über die CPU-Auslastung durch das Projekt können Sie über die „**cifX-Test Application**“ bekommen, sofern die netPLC-Komponente über den PCI-Bus oder Ethernet adressierbar ist.

- Starten Sie das Programm **cifX Test** aus der Systemsteuerung heraus.
- Stellen Sie die Verbindung zur netPLC-Komponente über **Device > Open** her.




- Wählen Sie **Information** ① > **Channel Information** ②.
- Wählen Sie **Status Block** ③ aus.
- Die Information zur CPU-Auslastung finden Sie in der mit ④ markierten **Zeile “System CPU Load**”.


© Hilscher, 2009 - 2014

# 13.4 Abbildungsverzeichnis

Abbildung 1: CoDeSys Hauptfenster 16
Abbildung 2: Neues Projekt 17
Abbildung 3: Standardprojekt, Geräteauswahl 18
Abbildung 4: Standardprojekt, Geräteübersicht 18
Abbildung 5: NPLC-C100 Host-E/A anhängen (1) 19
Abbildung 6: NPLC-C100 Host-E/A anhängen (2) 20
Abbildung 7: C100 Host-E/As konfigurieren (1) 20
Abbildung 8: C100 Host-E/As konfigurieren (2) 21
Abbildung 9: M100 Lokale E/As anhängen (1) 22
Abbildung 10: M100 Lokale E/As anhängen (2) 23
Abbildung 11: NPLC-M100 Lokale E/As konfigurieren 23
Abbildung 12: Feldbus auswählen 26
Abbildung 13: PROFIBUS Master NetX Konfiguration 27
Abbildung 14: PROFIBUS Master: DP-Parameter Konfiguration 27
Abbildung 15: CANopen 2. Dialog auswählen 28
Abbildung 16: CANopen Manager 29
Abbildung 17: NetX_CANbus > NetX Konfiguration 30
Abbildung 18: NetX_CANbus > CANbus 30
Abbildung 19: CANopen_Manager > CANopen_Manager 31
Abbildung 20: CANopen > Gerät anhängen 32
Abbildung 21: CANopen Manager 33
Abbildung 22: NetX_CANbus > NetX Konfiguration 34
Abbildung 23: NetX_CANbus > CANbus 34
Abbildung 24: Geräte-Feldbusadresse einstellen 35
Abbildung 25: Geräte-Beschreibungsdatei importieren 36
Abbildung 26: Slave Gerät anhängen 37
Abbildung 27: Slave Gerät auswählen 38
Abbildung 28: Slave Gerät eingefügt 38
Abbildung 29: Slave E/A-Adressen ermitteln 39
Abbildung 30: Slave Gerät eingefügt 40
Abbildung 31: Editor Symbols 44
Abbildung 32: Editor Symbols (2) 45
Abbildung 33: Editor Symbols Zugriffrechte 46
Abbildung 34: Meldungen nach Übersetzung 47
Abbildung 35: Kommunikationseinstellungen 48
Abbildung 36: Aktiven Pfad setzen 50
Abbildung 37: Pfad auf aktiv gesetzt 50
Abbildung 38: Meldungen zur Programmübersetzung 52
Abbildung 39: Gestartetes SPS-Programm 54
Abbildung 40: OPC-Server Konfigurationseinstieg 57
Abbildung 41: Host-E/A-Schnittstelle 62
Abbildung 42: Bibliothek hinzufügen 63
Abbildung 43: Bibliothek hinzufügen 64
Abbildung 44: Eingabehilfe Bausteinaufrufe 65
Abbildung 45: Systemfunktion StatusOfBattery 66
Abbildung 46: Systemfunktion StatusOfExt24VDC 67
Abbildung 47: Systemfunktion StatusOfPCIPower 68
Abbildung 48: Busdiagnose (1) Beispiel PROFIBUS 72
Abbildung 49: Busdiagnose (2) Beispiel PROFIBUS 72
Abbildung 50: Busdiagnose (3) Slave E/A-Daten 73
Abbildung 51: Object reference not set… 74
Abbildung 52: netPLC CoDeSys Server Tray Icon 85
Abbildung 53: netPLC CoDeSys Server Fenster 86
Abbildung 54: Speicherplatzbedarf des Projektes 91


© Hilscher, 2009 - 2014

# 13.5 Tabellenverzeichnis

Tabelle 1: Änderungsübersicht 5
Tabelle 2: Bezug auf netPLC-Komponenten (Hardware) 7
Tabelle 3: Bezug auf Firmware 7
Tabelle 4: Bezug auf Treiber 7
Tabelle 5: Bezug auf Server 7
Tabelle 6: Bezug auf Software 7
Tabelle 7: Dokumentationen netPLC CoDeSys 8
Tabelle 8: Systemfunktionen – CmpSysFunctions_netPLC 64
Tabelle 9: Systemfunktion StatusOfBattery 66
Tabelle 10: Systemfunktion StatusOfBattery, Hardware ohne AD-Wandler 67
Tabelle 11: Systemfunktion StatusOfExt24VDC 67
Tabelle 12: Systemfunktion StatusOfPCIPower 68
Tabelle 13: RUN/STOP-Betriebswahlschalter 69
Tabelle 14: Übersicht LEDs NPLC-C100 und NPLC-M100 und deren Bedeutung nach Protokoll (Feldbus-System) 75
Tabelle 15: Übersicht LEDs NHMI J500Q5-DP-DP/DPM/CDS/QV 75
Tabelle 16: Übersicht LEDs NHMI J500Q5-RE-DP/PNM/CDS/QV 76
Tabelle 17: System-LED 76
Tabelle 18: LED APL CoDeSys 77
Tabelle 19: LEDs DeviceNet-Master 77
Tabelle 20: LEDs PROFINET IO-RT-Controller 78
Tabelle 21: Fehlernummern Funktionen HilPlcSysFunctions_netPLC 79
Tabelle 22: Technische Daten CANopen-Master Protokoll 80
Tabelle 23: Technische Daten DeviceNet-Master Protokoll 81
Tabelle 24: Technische Daten PROFIBUS DP-Master Protokoll 82
Tabelle 25: Technische Daten PROFIBUS DP-Master Protokoll 83
Tabelle 26: Technische Daten PROFINET IO-Controller Protokoll 84
Tabelle 27: netPLC CoDeSys Server Fenster – Elemente 86
Tabelle 28: Beispiel – netPLC System Function 88
Tabelle 29: Beispiel – netPLC Open Process Control 88
Tabelle 30: Beispiel – netPLC Real Time Clock 89
Tabelle 31: Beispiel – NPLC-C100-DP Profibus Simple IO 89
Tabelle 32: Beispiel – NPLC-C100-DP with Host IOs 90
Tabelle 33: Beispiel – NPLC-C100-DP DPV1 90
Tabelle 34: Beispiel – NHMI J500Q5 90


© Hilscher, 2009 - 2014

## 13.6 Kontakte

### Hauptsitz

**Deutschland**
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Telefon: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

**Support**
Telefon: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

### Niederlassungen

**China**
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Telefon: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

**Support**
Telefon: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

**Frankreich**
Hilscher France S.a.r.l.
69500 Bron
Telefon: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

**Support**
Telefon: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

**Indien**
Hilscher India Pvt. Ltd.
New Delhi - 110 065
Telefon: +91 11 26915430
E-Mail: info@hilscher.in

**Italien**
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Telefon: +39 02 25007068
E-Mail: info@hilscher.it

**Support**
Telefon: +39 02 25007068
E-Mail: it.support@hilscher.com

**Japan**
Hilscher Japan KK
Tokyo, 160-0022
Telefon: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

**Support**
Telefon: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

**Korea**
Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Telefon: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

**Schweiz**
Hilscher Swiss GmbH
4500 Solothurn
Telefon: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

**Support**
Telefon: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

**USA**
Hilscher North America, Inc.
Lisle, IL 60532
Telefon: +1 630-505-5301
E-Mail: info@hilscher.us

**Support**
Telefon: +1 630-505-5301
**E-Mail:** us.support@hilscher.com


© Hilscher, 2009 - 2014